PRO-LOG® CORPORATION

STD 7000

7606
Parallel I/O Card

USER'S MANUAL

3/84

FOREWORD

This manual explains how to use Pro-Log's 7606 parallel I/O card. It is structured to reflect the answers to basic questions that you, the user, might ask yourself about the 7606. We welcome your suggestions on how we can improve our instructions.

The 7606 is part of Pro-Log's Series 7000 STD BUS hardware. Our products are modular, and they are designed and built with second-sourced parts that are industry standards. They provide the industrial manager with the means of using his own people to control the design, production, and maintenance of the company's products that use STD BUS hardware.

Pro-Log supports its products with thorough and complete documentation. Also, to provide maximum assistance to the user, we teach courses on how to design with, and to use, microprocessors and the STD BUS products.

You may find the following Pro-Log documents useful in your work: the **Microprocessor User's Guide** and the **Series 7000 STD BUS Technical Manual**. If you would like a copy of these publications, please submit your request on your company letterhead.

REFERENCES/BIBLIOGRAPHY

Series 7000 STD BUS Technical Manual, (Pro-Log Document #107855).

Microprocessor User's Guide, (Pro-Log Document #101145).

Zilog 1982/83 Data Book, Z80 PIO Parallel Input/Output Controller, Zilog Inc., 1983, pp. 67-79.

SGS Data Book Z80 Microprocessor Family, Z80 PIO Parallel Input/Output Controller, 3rd ed., pp. 91-106.

Mostek 1982/83 Z80 Designers Guide, MK3881 Parallel I/O Controller, Mostek Corporation, 1982, pp. 111-89 to 111-122.

Contents

	Page
Foreword	ii
References/Bibliography	ii
Tables	iii
Figures	iv
Section 1. Purpose and Main Features	1-1
Purpose	1-1
Features	1-1
Functional Description	1-2
Section 2. Installation and Specifications	2-1
Introduction	2-1
Installation	2-1
Jumper Configuration	2-2
I/O Port Addressing	2-2
Operational Modes	2-3
Handshake Signal Polarity	2-4
7606/STD BUS Interface	2-4
7606/User Interface	2-8
Port Data Bus and Handshake Signals	2-10
Structure	2-10
Termination Networks	2-10
Buffers	2-11
Specifications	2-11
Section 3. Operation and Programming	3-1
Introduction	3-1
Reset Status	3-1
Operational Modes	3-1
Byte Output	3-1
Byte Input	3-3
Byte Bidirectional	3-4
Bit Control	3-5
Programming	3-8
Interrupt Structure	3-10
Section 4. Operating Software	4-1
Introduction	4-1
Source Code Listings	4-4
Section 5. Maintenance	5-1
Reference Drawings	5-1
Return for Repair Procedures	5-5
Limited Warranty	5-5
Appendix A. Application	A-1

Tables

Table		Page
2-1	Sources for Shorting Connector	2-2
2-2	W1 Jumper Configuration for I/O Mapping	2-3
2-3	W2 Jumper Configuration for Operational Mode Selection	2-3
2-4	Port Handshake Options	2-4
2-5	STD Edge Connector Pin List	2-4
2-6	7606/User Interface Connector Pin List J1 and J2 Pin Configuration	2-8
2-7	Port Data Buffer Options	2-11
4-1	7606 Programmed Configuration for the Operating Software	4-1

Figures

Figure		Page
1-1	7606 Parallel Input/Output Card	1-1
1-2	Block Diagram	1-2
2-1	Installation	2-1
2-2	Jumper Configuration	2-2
2-3	Port Architecture	2-2
2-4	Output Timing	2-5
2-5	Input Timing	2-6
2-6	Interrupt Request/Acknowledge Timing	2-7
2-7	7606/User Interface Timing	2-9
2-8	Provisions for Termination Network	2-10
2-9	Termination Networks	2-10
3-1	Example of Mode 0 Configuration	3-1
3-2	Port Timing for Mode 0	3-2
3-3	Port Timing for Mode 0 with Ready Active at the Start of a Write Cycle	3-2
3-4	Example of Mode 1 Configuration	3-3
3-5	Port Timing for Mode 1	3-3
3-6	Port Timing for Mode 1 with Ready Active at the Start of a Read Cycle	3-4
3-7	Example of Mode 2 Configuration	3-4
3-8	Port A Timing for Mode 2	3-5
3-9	Examples of Mode 3 Operation	3-6
3-10	Port Timing for a CPU Read During Mode 3	3-7
3-11	Port Timing for Mode 3 Example	3-7
3-12	Mode Control Byte	3-8
3-13	Interrupt Vector Byte	3-9
3-14	I/O Register Control Byte	3-9
3-15	Interrupt Control Byte	3-9
3-16	Mask Control Byte	3-9
3-17	Interrupt Disable Byte	3-9
3-18	Vector Processing Sequence	3-10
3-19	Serial Priority Scheme	3-11
3-20	Daisy Chain Interrupt Servicing	3-12
3-21	Interrupt Request/Acknowledge Timing	3-13
3-22	Return from Interrupt Timing for a Port Currently Under Service	3-13
5-1	Schematic	5-2
5-2	Assembly Drawing	5-4
A-1	Example Using Bit Control (Mode 3)	A-1
A-2	Mode Control Byte	A-2
A-3	I/O Register Control Byte	A-2
A-4	Vector Control Byte	A-2
A-5	Interrupt Control Byte	A-2
A-6	Mask Control Byte	A-2
A-7	New Interrupt Control Byte	A-2

SECTION 1
Purpose and Main Features

PURPOSE

The 7606 is a highly versatile, four-port parallel input/output (PIO) card for Z80-based STD BUS systems. It provides a variety of cost-effective methods for data communications between the CPU and peripheral devices. The independent programmable ports allow four operating modes. Each port consists of eight data lines plus two handshake lines. Additionally, each port has interrupt capability and has a programmable vector for Z80 Mode-2 interrupts. The data lines have socketed buffers which can be replaced to obtain complemented signals. Handshake line polarity is also independently controlled by onboard jumpers. The 7606 has onboard interrupt logic to permit full use of the Z80's interrupt priority chain.

FEATURES

- Four operating modes
- Direct TTL interface
- Four independent I/O ports
- Interrupt driven data transfer
- Supports Z80 Mode-2 interrupts
- Automatic interrupt vectoring
- +5V operation
- STD Z80 BUS compatible

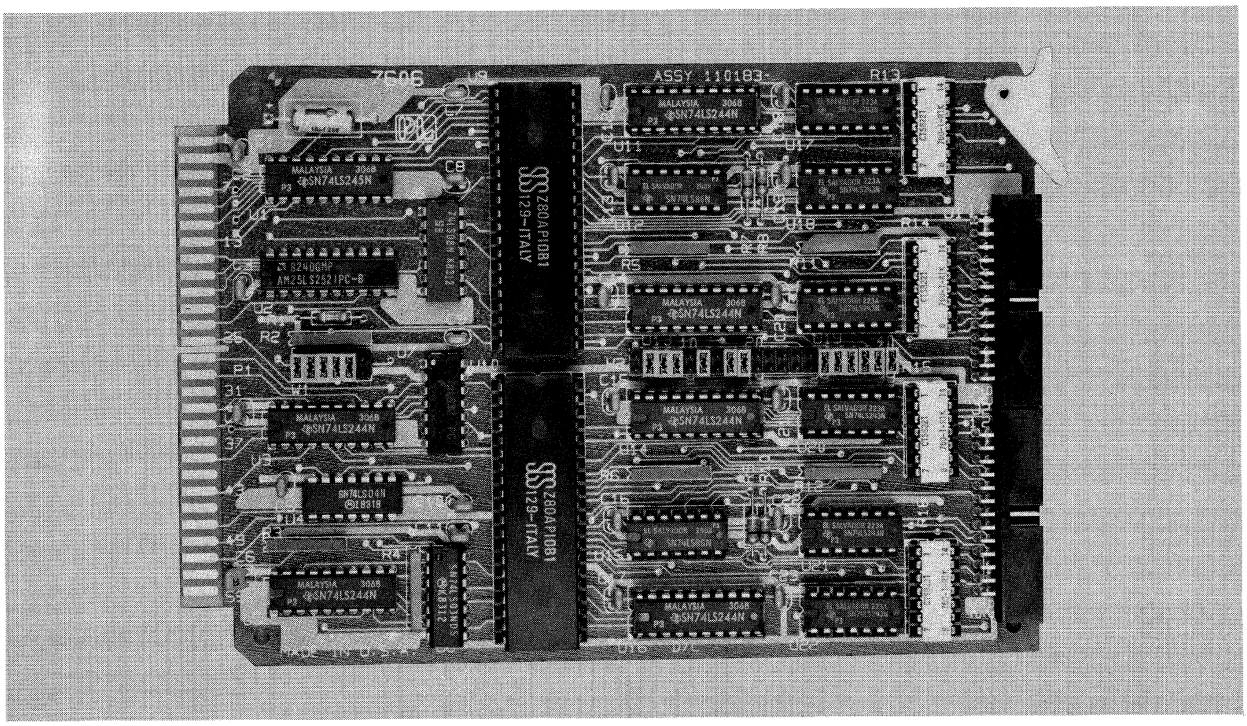


Figure 1-1. 7606 Parallel Input/Output Card

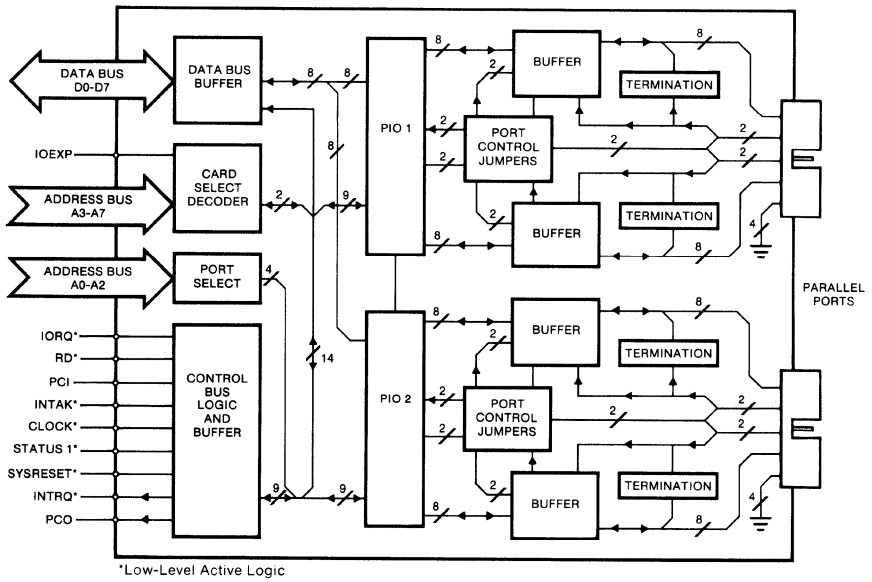


Figure 1-2. 7606 Block Diagram

FUNCTIONAL DESCRIPTION

The functional description defines each block in the 7606 block diagram (figure 1-2). The block diagram and the functional description are designed to familiarize the user with terms used throughout the manual.

Data Bus Buffer — an octal bus transceiver designed to provide buffering and direction control of the data bus.

Card Select Decoder — an address comparitor which provides a card select signal when any of the ports used by the card are addressed. The card select signal occurs when the logic state of the address lines is the same as the jumper selected card address.

Port Select Logic — if enabled by the card select decoder, the logic selects one of the eight possible port addresses on the card.

Control Bus Buffer and Logic — provides the control signal interface between the STD BUS and the card. Included in this interface is the direction control for the data bus buffer, priority chain look-ahead for the interrupt structure, and reset circuitry for the parallel input/output controllers.

PIO 1 and PIO 2 — the parallel input/output controllers which are the communication link between the STD BUS and user interface. PIO 1 and PIO 2 are identical in operation. They differ only in port addressing, interrupt vector, and priority in the interrupt structure.

Port Control Jumpers—a collection of unshrouded headers and shorting connectors, each block of which performs two functions. The first is to hardware configure the buffers into the selected mode of operation, and the second is to define the polarity of the handshake signals.

Buffers — increase the drive capability and the noise immunity of the signals available on user connectors J1 and J2. The buffers must be hardware configured to support the software selected operating mode.

Termination Networks — shipped as pull-up resistors, this is all that is required for most applications. The pull-up resistors are socketed to permit replacement with impedance matching networks (used when the port data bus and the strobe handshake signals are treated as transmission lines).

J1 and J2 — 26-pin right angle connectors which are the user interface to the card. Each connector contains all port data and handshake signals for a single PIO device.

SECTION 2
Installation and Specifications

INTRODUCTION

This section explains the procedure for installing the 7606 card into an STD BUS system. The installation includes options and specifications that must be considered before the card is installed.

INSTALLATION

> **CAUTION**
> To prevent possible damage to your STD BUS system make sure power is **off** before inserting a card into the card rack or before removing a card from the card rack.

The 7606 is designed to operate as part of an STD BUS system which must include a 2.5 or 4.0 MHz Z80 CPU-based processor card. The 7606 can be inserted into any slot in the STD BUS card rack. The ejector tab must be at the top as illustrated in figure 2-1. Power should be **off** during installation and removal of the card.

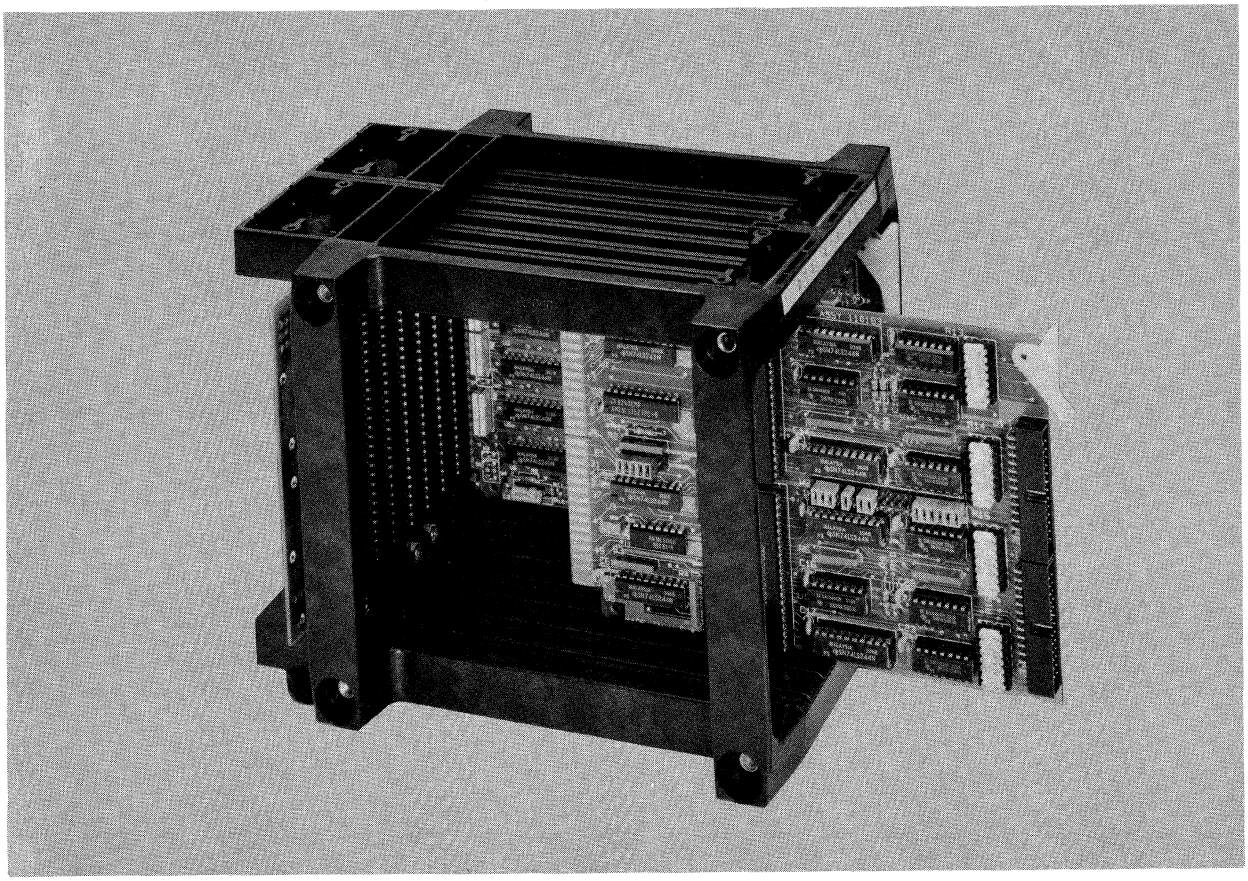


Figure 2-1. Installation (7606)

JUMPER CONFIGURATION

The 7606 has onboard jumpers to configure the card to meet system requirements. The jumper configurations consist of unshrouded headers and shorting connectors, as in figure 2-2. The shorting connectors can be removed and installed without the use of soldering equipment. Table 2-1 lists sources for the shorting connectors.

The jumper configurations are divided into three functional groups: I/O Port Addressing, Operational Modes, and Handshake Signals.

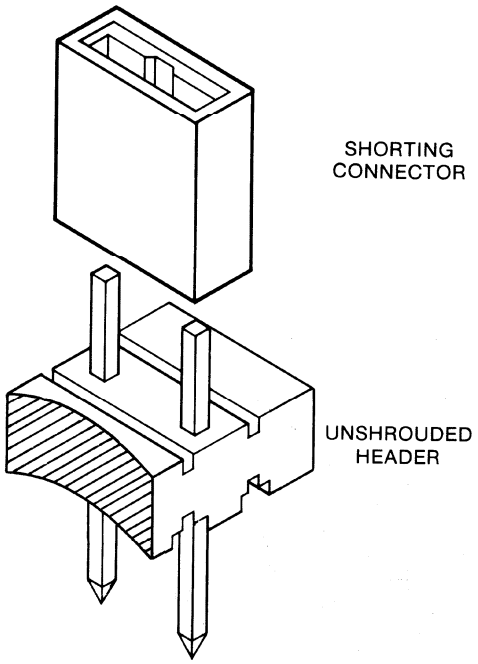


Figure 2-2. Jumper Configuration (7606)

SOURCE	PART NUMBER
PRO-LOG	902520
ELCO	008261024200870
AMP	531220-3

Table 2-1. Sources for Shorting Connector

I/O Port Addressing

All communications between the 7606 and the STD BUS are through I/O ports. The 7606 port architecture is illustrated in figure 2-3. As seen in figure 2-3, there are four ports, each of which occupies two addresses. One port address is used to enter control bytes and another to transfer data. The 7606 occupies a total of eight sequential port addresses.

Table 2-2 illustrates the jumper options for mapping the 7606 I/O ports. The table shows the starting address for each block of eight sequential ports required by the 7606. The IOEXP jumper (pins 1 and 2) is shown installed for all possible mapping locations. This jumper supports the IOEXP signal for the STD BUS. Pro-Log's CPU cards tie this signal to ground. Refer to the **STD BUS Technical Manual** for more information on IOEXP.

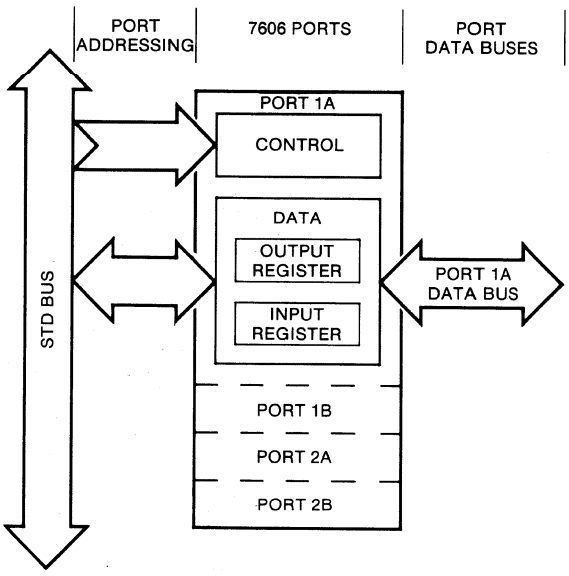


Figure 2-3. Port Architecture (7606)

ADDRESS	W1 JUMPER 1-2	3-4	5-6	7-8	9-10	11-12
00	X	X	X	X	X	X
*08	X	X	X	X	X	
10	X	X	X	X		X
18	X	X	X	X		
20	X	X	X		X	X
28	X	X	X		X	
30	X	X	X			X
38	X	X	X			
40	X	X		X	X	X
48	X	X		X	X	
50	X	X		X		X
58	X	X		X		
60	X	X			X	X
68	X	X			X	
70	X	X				X
78	X	X				
80	X		X	X	X	X
88	X		X	X	X	
90	X		X	X		X
98	X		X	X		
A0	X		X		X	X
A8	X		X		X	
B0	X		X			X
B8	X		X			
C0	X			X	X	X
C8	X			X	X	
D0	X			X		X
D8	X			X		
E0	X				X	X
E8	X				X	
F0	X					X
F8	X					

* As shipped
X Jumper installed

Table 2-2. W1 Jumper Configuration for I/O Mapping (7606)

Operational Modes

The 7606 card supports four modes of operation: Byte Output, Byte Input, Byte Bidirectional, and Bit Control. To operate a port in one of these modes, the port must be hardware configured with jumpers, and software configured with control bytes. An explanation of each mode of operation and the control bytes is presented in section 3. This section illustrates the procedure for using jumpers to hardware configure a port for a mode of operation.

There are four ports on the 7606. The ports are labeled 1A, 1B, 2A, and 2B, each of which may be configured to a unique mode of operation, except that ports 1B and 2B do not support the byte bidirectional mode.

Table 2-3 lists each port and the jumper configuration for all possible modes of operation. Each jumper is referenced by a pair of pin numbers. The configuration as shipped is also found in table 2-3.

PORT	MODE	OPERATION	OPEN JUMPERS	SHORTED JUMPERS
1A	0	Byte Output	17-18, 25-26, 27-28	21-22
	1	Byte Input	21-22, 25-26, 27-28	17-18[4]
	2	Bidirectional	17-18, 21-22	25-26, 27-28
	3	Bit Control	[1]	[1]
1B	0	Byte Output	—	33-34, 37-38[4]
	1	Byte Input	33-34, 37-38	—
	2	Bidirectional	[2]	[2]
	3	[3]Bit Control		
		MSB Output	—	37-38
		MSB Input	37-38	—
		LSB Output	—	33-34
		LSB Input	33-34	—
2A	0	Byte Output	19-20, 29-30, 31-32	23-24
	1	Byte Input	23-24, 29-30, 31-32	19-20
	2	Bidirectional	19-20, 23-24	29-30, 31-32[4]
	3	Bit Control	[1]	[1]
2B	0	Byte Output	—	35-36, 39-40[4]
	1	Byte Input	35-36, 39-40	—
	2	Bidirectional	[2]	[2]
	3	[3]Bit Control		
		MSB Output	—	35-36
		MSB Input	35-36	—
		LSB Output	—	39-40
		LSB Input	39-40	—

[1]Same as Byte Input or Byte Output jumper configuration.
[2] Not available.
[3] Adjustable in 4-bit sections. MSB=Most Significant Byte; LSB=Least Significant Byte.
[4] As shipped.

Table 2-3. W2 Jumper Configuration for Operational Mode Selection (7606)

Handshake Signal Polarity

The 7606 provides a parallel interface between a Z80-based STD BUS and peripheral devices. The interface contains handshaking signals that control the data transfer. Each port contains two handshake signals: Ready and Strobe. A description of these signals is found in section 3. This section shows how to define the polarity of these signals with jumpers. The ready (RDY) signal is active high in its noninverted state. The strobe (STB) signal is active low in its noninverted state. Table 2-4 illustrates each handshake signal and the jumper configuration for its inverted and noninverted polarities.

PORT	HANDSHAKE SIGNAL	LOGIC	W2 JUMPER CONFIGURATION
1A	RDY	*Inverted	(1,2) Open
		Noninverted	(1,2) Shorted
	STB	*Inverted	(13,14) Open
		Noninverted	(13,14) Shorted
1B	RDY	*Inverted	(9,10) Open
		Noninverted	(9,10) Shorted
	STB	Inverted	(5,6) Open
		*Noninverted	(5,6) Shorted
2A	RDY	Inverted	(3,4) Open
		*Noninverted	(3,4) Shorted
	STB	Inverted	(15,16) Open
		*Noninverted	(15,16) Shorted
2B	RDY	Inverted	(11,12) Open
		*Noninverted	(11,12) Shorted
	STB	Inverted	(7,8) Open
		*Noninverted	(7,8) Shorted

*As shipped

Table 2-4. Port Handshake Options

7606/STD BUS INTERFACE

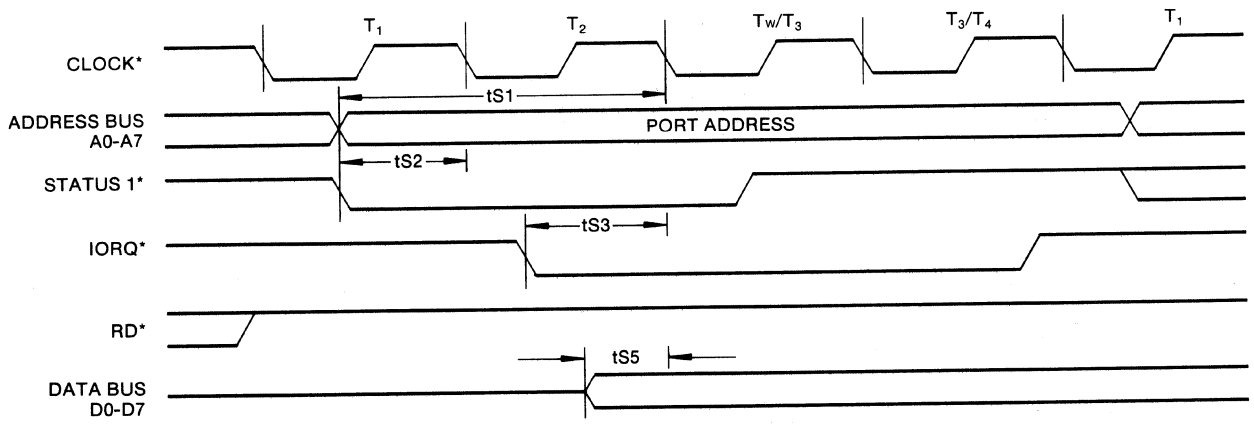
The 7606 card is designed for use with the STD BUS system. The STD BUS defines the mechanical and electrical specifications of the cards using it. The electrical signals defined by the STD BUS are listed in table 2-5. The 7606/STD BUS interface signals are shown with their drive capabilities.

PIN NUMBER				PIN NUMBER			
OUTPUT (LSTTL DRIVE)					OUTPUT (LSTTL DRIVE)		
INPUT (LSTTL LOADS)						INPUT (LSTTL LOADS)	
MNEMONIC							MNEMONIC
+5V	IN		2	1		IN	+5V
GROUND	IN		4	3		IN	GROUND
-5V			6	5			-5V
D7	1	55	8	7	55	1	D3
D6	1	55	10	9	55	1	D2
D5	1	55	12	11	55	1	D1
D4	1	55	14	13	55	1	D0
A15			16	15		1	A7
A14			18	17		1	A6
A13			20	19		1	A5
A12			22	21		1	A4
A11			24	23		1	A3
A10			26	25		1	A2
A9			28	27		1	A1
A8			30	29		1	A0
RD*	1		32	31			WR*
MEMRQ*			34	33		1	IORQ*
MEMEX			36	35		1	IOEXP (GROUND)
MCSYNC*			38	37			REFRESH*
STATUS 0*			40	39		1	STATUS 1* (M1*)
BUSRQ*			42	41			BUSAK*
INTRQ* (INT*)		18	44	43		1	INTAK*
NMIRQ* (NMI*)			46	45			WAITRQ* (WAIT*)
PBRESET*			48	47		1	SYSRESET*
CNTRL* (EXT CLK IN)			50	49		1	CLOCK*
PCI	15		52	51	18		PCO
AUX GND			54	53			AUX GND
AUX -V			56	55			AUX +V

*Low-Level Active Logic
( ) Designates equivalent Z80A pin names

Table 2-5. STD Edge Connector (7606)

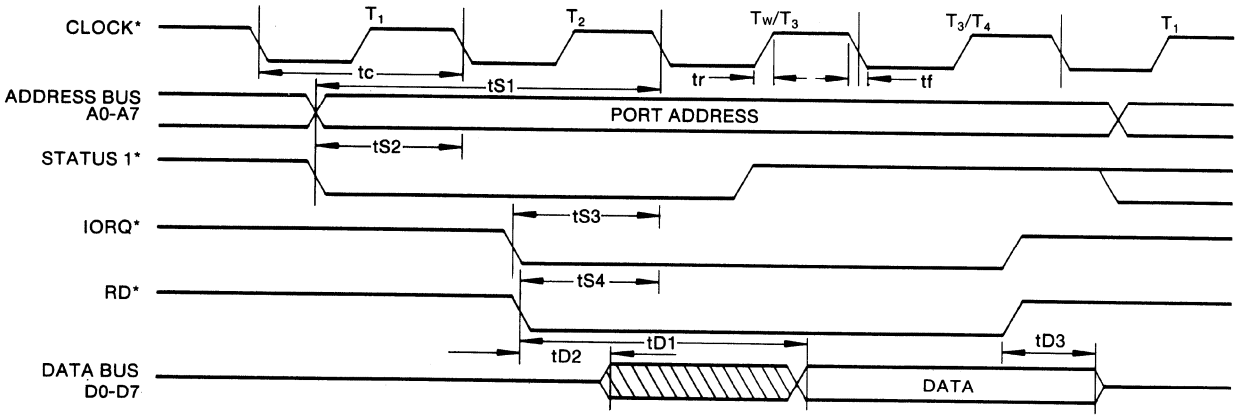
The timing requirements of the 7606 for the STD BUS interface are shown in figures 2-4 through 2-6. The timing has been divided into three functional groups: Output Timing (figure 2-4), Input Timing (figure 2-5), and Interrupt Request/Acknowledge Timing (figure 2-6).



		2.5MHz		4.0 MHz	
SYMBOL	PARAMETER	MIN	MAX	MIN	MAX
tS1	Address setup time to falling edge of CLOCK*	335		195	
tS2	STATUS1* setup time to falling edge of CLOCK*	240		120	
tS3	IORQ* setup time to falling edge of CLOCK*	320		185	
tS5	Data setup time to falling edge of CLOCK*	60		60	

All times in nanoseconds.

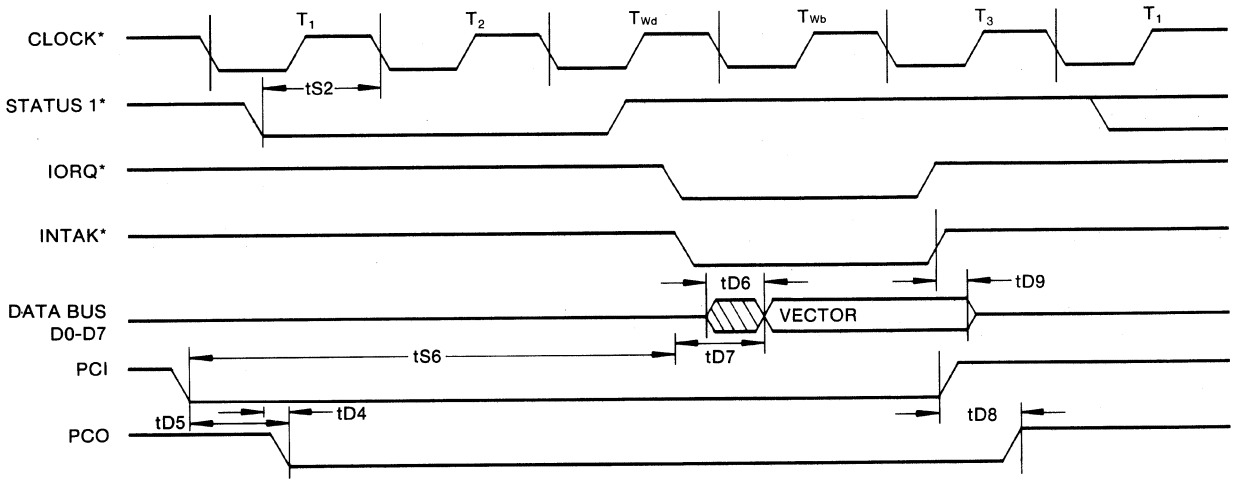
Figure 2-4. Output Timing (7606)



SYMBOL	PARAMETER	2.5MHz MIN	2.5MHz MAX	4.0 MHz MIN	4.0 MHz MAX
tD1	Data delay from falling edge of RD*		470		
tD2	Data bus buffer direction switching delay from falling edge of IORQ*		90		90
tD3	Floating bus delay from rising edge of CLOCK*		200		150
tS1	Address setup time to falling edge of CLOCK*	335		195	
tS2	STATUS1* setup time to falling edge of CLOCK*	240		120	
tS3	IORQ* setup time to falling edge of CLOCK*	320		185	
tS4	RD* setup time to falling edge of CLOCK*	250		125	
tc	CLOCK* period	400	(1)	250	(2)
tr	CLOCK* rise time		30		30
tf	CLOCK* fall time		30		30

All times in nanoseconds.
(1) 250 + tr + tf
(2) 400 + tr + tf

Figure 2-5. Input Timing (7606)



SYMBOL	PARAMETER	2.5MHz MIN	2.5MHz MAX	4.0 MHz MIN	4.0 MHz MAX
tD4	PCO delay from falling edge of STATUS1* (interrupt occurring prior to M1)	380		270	
tD5	PCO delay from falling edge of PCI		40		40
	PCO delay from falling edge of IEI2		40		40
tD6	Data bus buffer direction switching delay from falling edge of INTAK*		65		65
tD7	Data (vector) output delay from falling edge of IORQ*		380		290
tD8	PCO delay from rising edge of PCI		35		35
	PCO delay from rising edge of IEI2		30		30
tD9	Floating bus delay from rising edge of INTAK*		65		65
tS2	See figure 2-4	115		235	
tS6	PCI setup time to falling edge of IORQ*	150		150	

All times in nanoseconds.

Figure 2-6. Interrupt Request/Acknowledge Timing (7606)

7606/USER INTERFACE

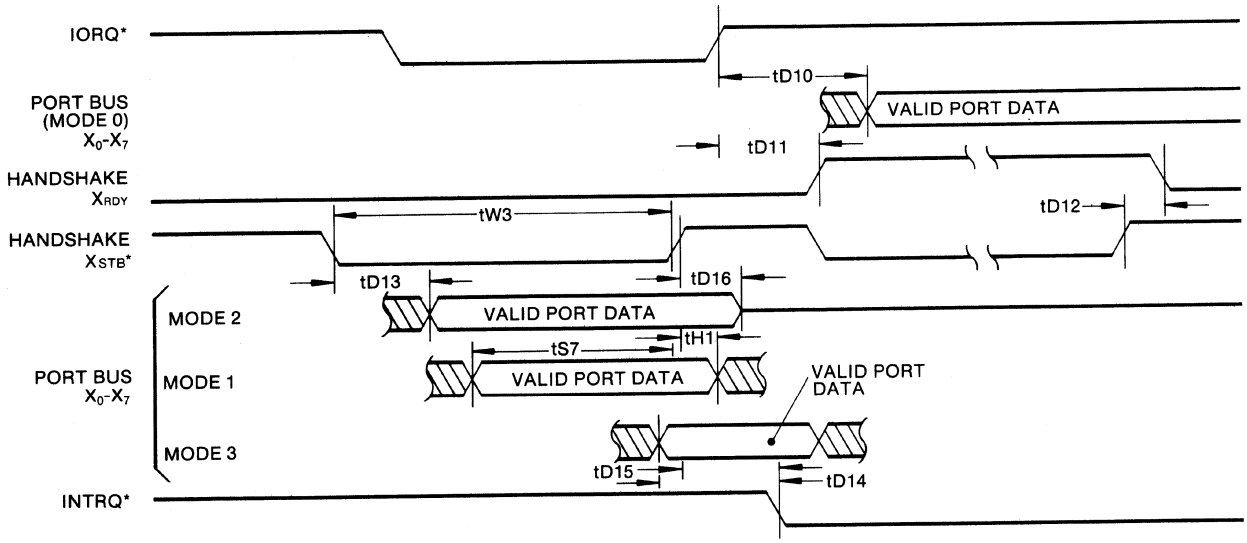
The 7606 provides two 26-pin header connectors, J1 and J2, for user interface.Each connector supports two independent ports and port associated handshake signals.Table 2-6 illustrates the pin configuration and the drive capability of each connector.

The timing for the signals available at J1 and J2 is illustrated in figure 2-7. The timing diagram shows the handshake signals with ready active high and strobe active low. Since the polarity of the handshake signal is jumper selectable, all possibilities are listed in figure 2-7.

J1 PIN NUMBER				J2 PIN NUMBER			
OUTPUT (LSTTL DRIVE)						OUTPUT (LSTTL DRIVE)	
INPUT (LSTTL LOADS)						INPUT (LSTTL LOADS)	
MNEMONIC PIO #1							MNEMONIC PIO #2
1ARDY		18	1	1	18		2ARDY
GND		OUT	2	2	OUT		GND
1ASTB*	14(1)		3	3	14(1)		2ASTB*
GND		OUT	4	4	OUT		GND
1A7	14(1)	45(55)	5	5	45(55)	14(1)	2A7
1A6	14(1)	45(55)	6	6	45(55)	14(1)	2A6
1A5	14(1)	45(55)	7	7	45(55)	14(1)	2A5
1A4	14(1)	45(55)	8	8	45(55)	14(1)	2A4
1A3	14(1)	45(55)	9	9	45(55)	14(1)	2A3
1A2	14(1)	45(55)	10	10	45(55)	14(1)	2A2
1A1	14(1)	45(55)	11	11	45(55)	14(1)	2A1
1A0	14(1)	45(55)	12	12	45(55)	14(1)	2A0
1B0	14(1)	45(55)	13	13	45(55)	14(1)	2B0
1B1	14(1)	45(55)	14	14	45(55)	14(1)	2B1
1B2	14(1)	45(55)	15	15	45(55)	14(1)	2B2
1B3	14(1)	45(55)	16	16	45(55)	14(1)	2B3
1B4	14(1)	45(55)	17	17	45(55)	14(1)	2B4
1B5	14(1)	45(55)	18	18	45(55)	14(1)	2B5
1B6	14(1)	45(55)	19	19	45(55)	14(1)	2B6
1B7	14(1)	45(55)	20	20	45(55)	14(1)	2B7
1BSTB*	14(1)		21	21		14(1)	2BSTB*
GND		OUT	22	22	OUT		GND
1BRDY		18	23	23	18		2BRDY
GND		OUT	24	24	OUT		GND
N/C			25	25			N/C
N/C			26	26			N/C

*Low-level active logic.
NOTE: The numbers in parentheses can be achieved with removal of termination.

Table 2-6. 7606/User Interface Connector Pin List
J1 and J2 Pin Configuration



SYMBOL	PARAMETER	2.5 MHz MIN	2.5 MHz MAX	4.0 MHz MIN	4.0 MHz MAX
tD10	Port data delay from rising edge of IORQ* (Mode 0)		240		220
tD11	XRDY delay from rising edge of IORQ*		tc+505		tc+455
	XRDY* delay from rising edge of IORQ*		tc+500		tc+450
tD12	XRDY delay from rising edge of XSTB*		tc+465		tc+425
	XRDY* delay from rising edge of XSTB*		tc+465		tc+425
	XRDY delay from falling edge of XSTB		tc+475		tc+435
	XRDY* delay from falling edge of XSTB		tc+470		tc+430
tD13	Port data delay from falling edge of XSTB* (Mode 2)		290		270
	Port data delay from rising edge of XSTB (Mode 2)				
tD14	INTRQ* delay from data match (Mode 3)		485		445
tD15	INTRQ* delay from rising edge of XSTB*		575		525
	INTRQ* delay from falling edge of XSTB		575		535
tD16	Floating port bus delay from rising edge of XSTB* (Mode 2)		240		260
	Floating port bus delay from falling edge of XSTB (Mode 2)		250		270
tH1	Port data hold time from rising edge of XSTB* (Mode 1)		250		230
	Port data hold time from falling edge of XSTB (Mode 1)	0		0	
tS7	Port data setup time to rising edge of XSTB* (Mode 1)	270		240	
	Port data setup time to falling edge of XSTB (Mode 1)	265		235	
tW3	XSTB* pulse width	150		150	

All times in nanoseconds.

Figure 2-7. 7606/User Interface Timing

PORT DATA BUS AND HANDSHAKE SIGNALS

Structure

There are two header connectors provided for the user interface to the 7606 which were discussed earlier in this section. Each connector provides two ports to the user. J1 provides ports 1A and 1B while J2 provides ports 2A and 2B. Ports 1A and 2A are functionally identical, as are ports 1B and 2B.

Each port on the 7606 is brought to the header connectors with a port data bus and a pair of handshake signals. To make the user interface versatile, several options for port configuration are available. The mode of operation and the polarity of the handshake signals are jumper selectable, as discussed earlier in this section. The 7606 also provides a means for the user to define the termination network and the polarity of the port data bus.

Termination Networks

Each port data bus and handshake signal has provisions for termination as illustrated in figure 2-8. The ready handshake line is terminated with a fixed series resistance. As shipped, the strobe handshake line and port data bus are terminated with 1K pull-up resistors as illustrated in figure 2-9(A).

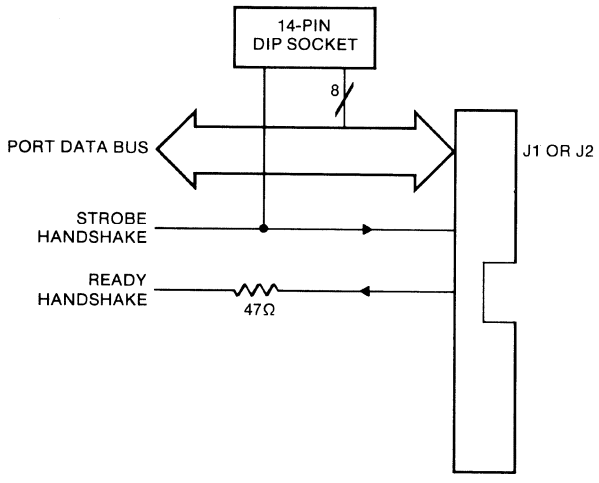
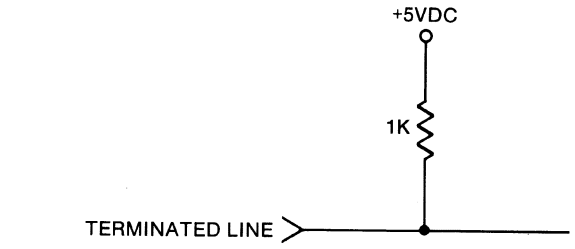


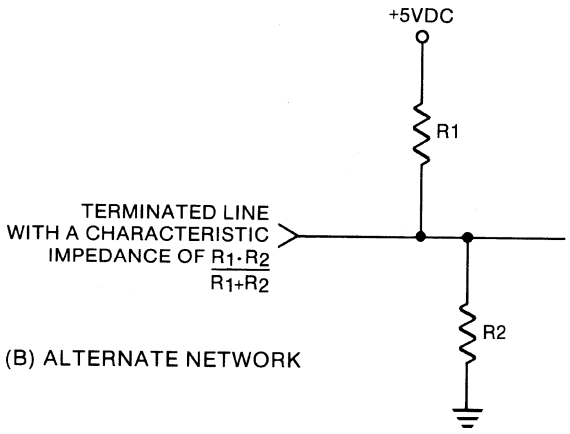
Figure 2-8. Provisions for Termination Network (7606)

The 7606 interfaces a wide variety of peripherals to the STD BUS. For more of these applications the passive pull-up resistors are adequate termination. If greater demands are placed on the signal path between the 7606 and the peripheral device it may be necessary to treat the signal lines as transmission lines.

A transmission line should be terminated in a resistance that is equal to its characteristic impedance. Proper termination eliminates signal reflection and standing waves. A substitute passive termination network is illustrated in figure 2-9(B). Once the characteristic impedance of the transmission line is known, the value of R1 and R2 can be determined. The manufacturers listed in figure 2-9(B) have a variety of networks available.



(A) AS SHIPPED NETWORK



(B) ALTERNATE NETWORK

ALTERNATE TERMINATION NETWORK SOURCE
BOURNS 4114R-003 SERIES
CTS 760-5-R SERIES

Figure 2-9. Termination Networks (7606)

Buffers

The 7606 provides socketed buffers on each port data bus. Table 2-7 illustrates the port buffer options.

Port A data bus lines are buffered using two quad noninverting transceivers, allowing true bidirectional capability. These buffers are socketed to permit replacement of the existing transceivers with their logical complements.

Port B data bus lines are buffered to permit the direction of the bus to be defined in 4-bit segments. This means that port B can be 8 bits of output, 8 bits of input, or 4 bits of each. These buffers are socketed to permit replacement of the existing transceivers with their logical complements.

PORTS	NONINVERTING LOGIC	INVERTING LOGIC
A	*74LS244	74LS240
B	*74LS243	74LS242

*As shipped

Table 2-7. Port Data Buffer Options (7606)

Mechanical Specifications

The 7606 meets all the general STD BUS mechanical specifications.

SPECIFICATIONS

7606 Environmental Specifications

			Recommended Operating Limits			Absolute Non-Operating Limits	
MNEM	PARAMETER	UNIT	MIN	TYP	MAX	MIN	MAX
—	Free Air Temperature	°C	0	+25	+65	-40	+75
—	Noncondensing Relative Humidity	% RH	5	—	95	5	95

7606 Electrical Specifications

			Recommended Operating Limits			Absolute Non-Operating Limits	
MNEM	PARAMETER	UNIT	MIN	TYP	MAX	MIN	MAX
VCC	Supply Voltage	V	4.75	5.00	5.25	0	7.00

7606 Electrical Specifications Over Operating Range

			Recommended Operating Limits		
MNEM	PARAMETER	UNIT	MIN	TYP	MAX
ICC[1]	VCC Supply Current	mA		850	1100
	STD BUS Input Load—See table 2-5				
	STD BUS Output Load—See table 2-5				

[1]As shipped with 1K pullup networks.

SECTION 3
Operation and Programming

INTRODUCTION

The 7606 is a programmable quad port device providing an interface between a Z80 CPU and peripheral devices. The interface is accomplished under one of four modes of operation. The 7606 must be hardware configured and software programmed to support a selected mode. The hardware configuration is discussed in section 2. Section 3 is dedicated to the operation of each mode and the programming sequence used to select it.

RESET STATUS

The 7606 automatically enters a reset state when power is applied. The reset state performs the following functions:

A. The port data bus lines are pulled up to a logical 1 at the user interface.

B. Ready handshake signals are placed in the inactive state.This is a logical 0 for a noninverted signal, and a logical 1 for an inverted signal.

C. Mode 1 operation (not to be confused with Z80 Mode-1 interrupts) is automatically selected.

D. Interrupt requests are reset.

E. Port output registers are reset.

F. Vector address registers are not reset.

In addition to the power-on reset, the 7606 can be reset by placing SYSRESET* in its active state. Once the 7606 has entered a reset state it is held there until it receives a control word from the CPU. See **Programming** which follows in this section.

OPERATIONAL MODES

The four possible modes of operation are: Byte Output (Mode 0), Byte Input (Mode 1), Byte Bidirectional (Mode 2), and Bit Control (Mode 3).

Byte Output (Mode 0)

Any of the 7606 ports can be programmed to operate in Mode 0. Each port contains an output register that is individually addressed by the CPU. Data can be written to the port at any time with a CPU I/O write operation. Data can also be read from the port at any time with a CPU I/O read operation. Figure 3-1 illustrates a 7606 port in an output configuration.

When data is written to the port from the CPU, an active Ready output indicates to the external device that data is available. The external device then reads the data from the 7606 port. An active Strobe from the external device tells the 7606 that the data has been received. The strobe signal will reset the ready signal and generates an interrupt request if enabled.

The timing for Mode 0 is illustrated in figure 3-2. This timing diagram is used to explain the Mode 0 operation. For more precise specifications, refer to figure 2-7.

An output cycle is initiated with the execution of an I/O write instruction by the CPU. The 7606 internally generates a write signal (IWR*). An active IWR* signal is used to latch the data from the CPU data bus into the output register of the selected port on the 7606.

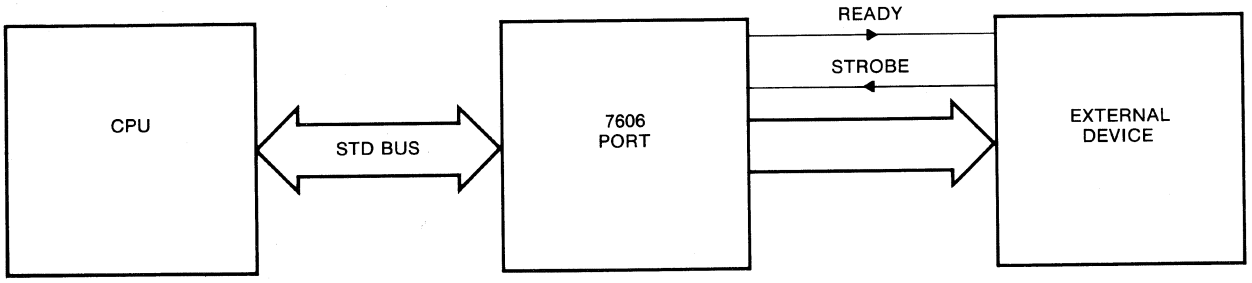


Figure 3-1. Example of Mode 0 Configuration (7606)

The rising edge of the IWR* pulse raises the ready line after the next falling edge of Clock. The ready signal can be used by the external device to latch the data available on the port data bus. The ready signal remains active until the 7606 is strobed by the external device. The transition of Strobe* from its active to its inactive state will generate an interrupt, if enabled, and reset the ready signal on the next falling edge of Clock.

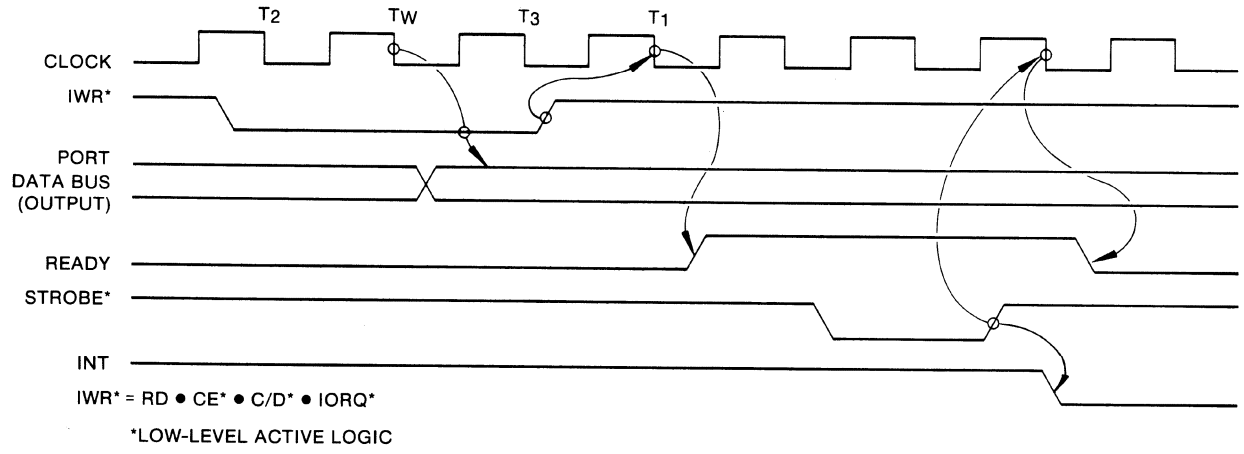


Figure 3-2. Port Timing for Mode 0 (7606)

It is possible for the CPU to perform an I/O write operation to a port whose external device has not yet transferred the data of a previous I/O write operation. In this case, the 7606 forces the ready signal into an inactive state during the time that the CPU is writing new data to the port output register. This is illustrated in figure 3-3 and is done to guarantee that the ready signal is inactive when the port data is being changed and that the ready signal generates a transition for edge latching external devices.

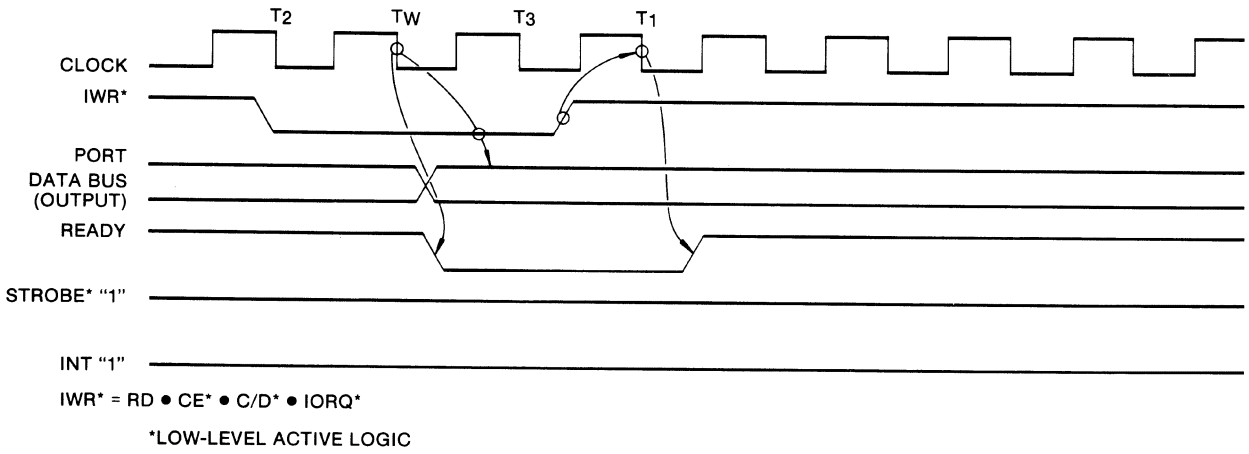


Figure 3-3. Port Timing for Mode 0 with Ready Active at the Start of a Write Cycle (7606)

If the 7606 is not in a reset status (a control mode has been selected), the output register may be loaded before Mode 0 is selected. This allows the port data bus to become active in a known state.

Byte Input (Mode 1)

Any of the 7606 ports can be programmed to operate in Mode 1. Each port contains an input register that is individually addressed by the CPU. Figure 3-4 illustrates a 7606 port in an input configuration.

When the CPU performs an I/O read from a 7606 port the ready signal goes active to indicate to the external device that the CPU is requesting data. The external device then places data on the port data bus and strobes it into the port input register. The strobe latches the data into the port input register, resets the ready signal, and generates an interrupt request if enabled.

The timing for Mode 1 is illustrated in figure 3-5. This timing diagram is used to explain the Mode 1 operation. For more precise specifications refer to figure 2-7. The timing diagram of figure 3-5 assumes the CPU has performed an I/O read operation that caused the 7606 to activate the ready signal.

The external device places Strobe* into its active state which loads data into the selected port input register of the 7606. The transition of Strobe* from an active to inactive state generates an interrupt request if enabled. The next falling edge of Clock resets Ready to its inactive state, indicating the input register is full and cannot accept more data until the CPU completes an I/O read.

The 7606 internally generates a read signal (IRD*) in response to a CPU I/O read operation. The positive edge of IRD* sets Ready into an active state on the next falling edge of Clock. At this time new data can be loaded into the port input register of the 7606.

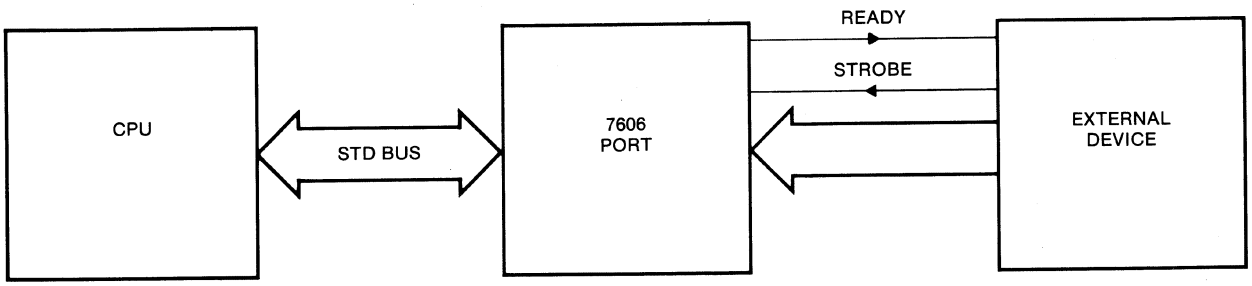


Figure 3-4. Example of Mode 1 Configuration (7606)

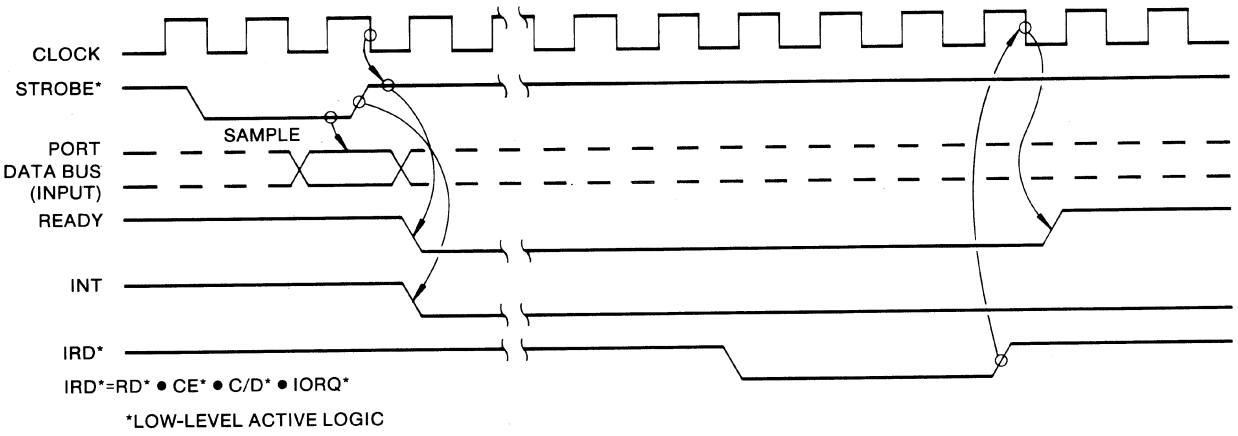


Figure 3-5. Port Timing for Mode 1 (7606)

The CPU can perform an I/O read operation from a port whose external device has not yet strobed data in. In this case the 7606 forces the active ready signal into an inactive state during the time the CPU is reading data from the port input register (see figure 3-6). If the external device strobes data into the 7606 only when Ready is high, the forced state of Ready prevents the port input register data from changing while the CPU is performing an I/O read. The ready signal returns active after the CPU reads the data.

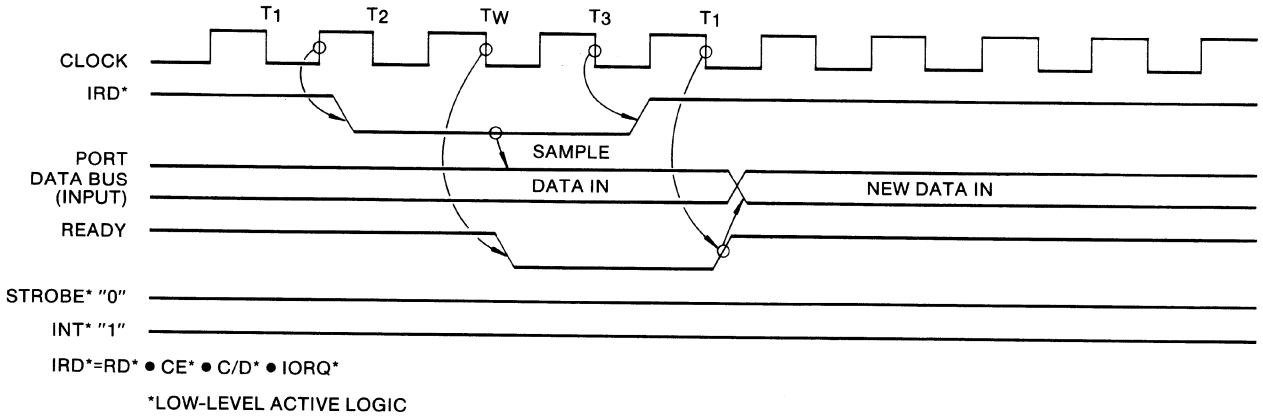


Figure 3-6. Port Timing for Mode 1 with Ready Active at the Start of a Read Cycle (7606)

Byte Bidirectional (Mode 2)

Only ports 1A and 2A can be programmed to operate in Mode 2. Mode 2 operation uses the 1A or 2A port data bus and handshake signals, as well as the port 1B or 2B interrupts and handshake signals. Port B must be set to Mode 3 and masked. Masking port B prevents it from generating an interrupt. In operation, port 1A or 2A is used for both data input and data output. Output operation is similar to Mode 0 except that data is not available on the port data bus until the 7606 receives an active strobe. Input operation is similar to Mode 1, except that handshake signals and interrupts are used from port 1B (if using port 1A) or port 2B (if using port 2A). Figure 3-7 illustrates a 7606 port A in a bidirectional configuration.

The timing for Mode 2 is illustrated in figure 3-8. This timing diagram is used to explain Mode 2 operation. For more precise specifications refer to figure 2-7.

The bidirectional mode combines the port A data bus and ports A and B handshake signals and interrupts. The port A handshake signals and interrupting capabilities are used for output operations. The port B handshake signals and interrupting capabilities are used for input operations. Data is allowed on the port data bus only when the A-strobe is active. The active to inactive state of Strobe* can be used to latch the data into the external device.

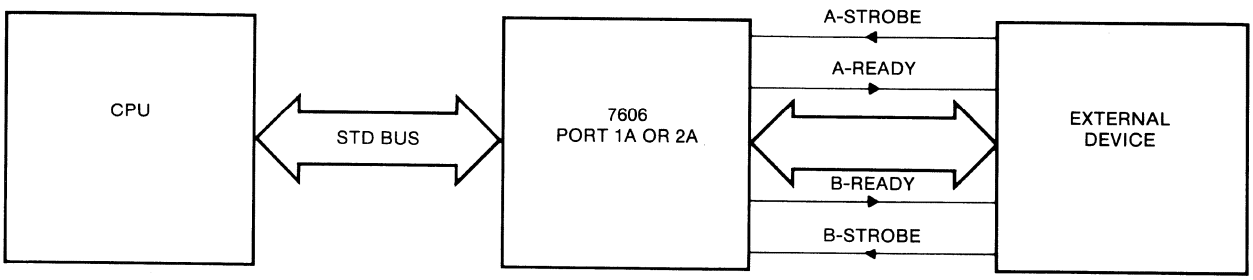


Figure 3-7. Example of Mode 2 Configuration (7606)

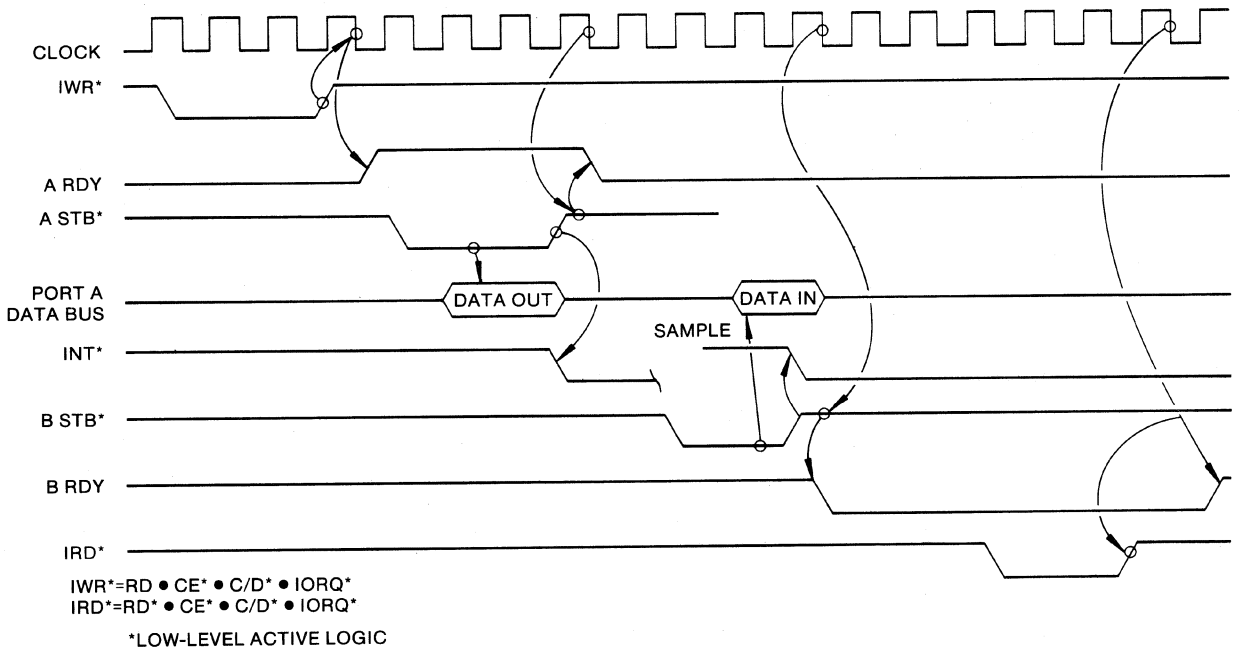


Figure 3-8. Port A Timing for Mode 2 (7606)

Bit Control (Mode 3)

Any of the 7606 ports can be programmed to operate in Mode 3. Each port contains an input and output register that work together during this mode of operation. The CPU can perform an I/O read or write at any time. The port data bus can be configured as outputs or inputs in 4-bit segments and individual unused bits can be masked. Figure 3-9 illustrates the possible configurations for a 7606 port operating in Mode 3.

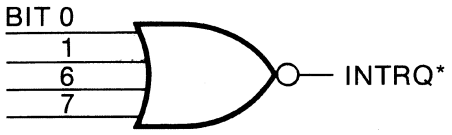
Mode 3 is intended for status and control operations and does not use the data transfer handshake signals. Instead, the port is programmed to generate an interrupt request on a specific condition of the unmasked signals in the port data bus.

The CPU can execute an I/O read or an I/O write at any time. When writing data to a port configured for Mode 3, the data is latched with the same timing as Mode 0. Figure 3-10 illustrates the timing for a CPU I/O read operation. This timing diagram is used to explain the Mode 3 operation. For more precise specifications refer to figure 2-7.

During an I/O read, the data returned to the CPU is composed of port output register data from output lines and port input register data from input lines. The input register contains data that was present immediately prior to the falling edge of the STD BUS RD* signal.

A logical equation is defined during the Mode 3 programming operation. The logical equation consists of a function (AND/OR), an active level (high/low), and masking signals not to generate an interrupt request. As an example, assume that a port on the 7606 has been programmed as follows:

- Mode 3 operation
- Interrupt enabled
- Logical OR function
- Active high
- Bits 2-5 masked



Under these conditions, if one unmasked line of the port data bus becomes active (Logical 1), an interrupt request is generated. Assume the same programming sequence is specified except the logical AND is specified instead of the logical OR. Under these conditions, if all unmasked lines of the port data bus become active high, an interrupt request is generated.

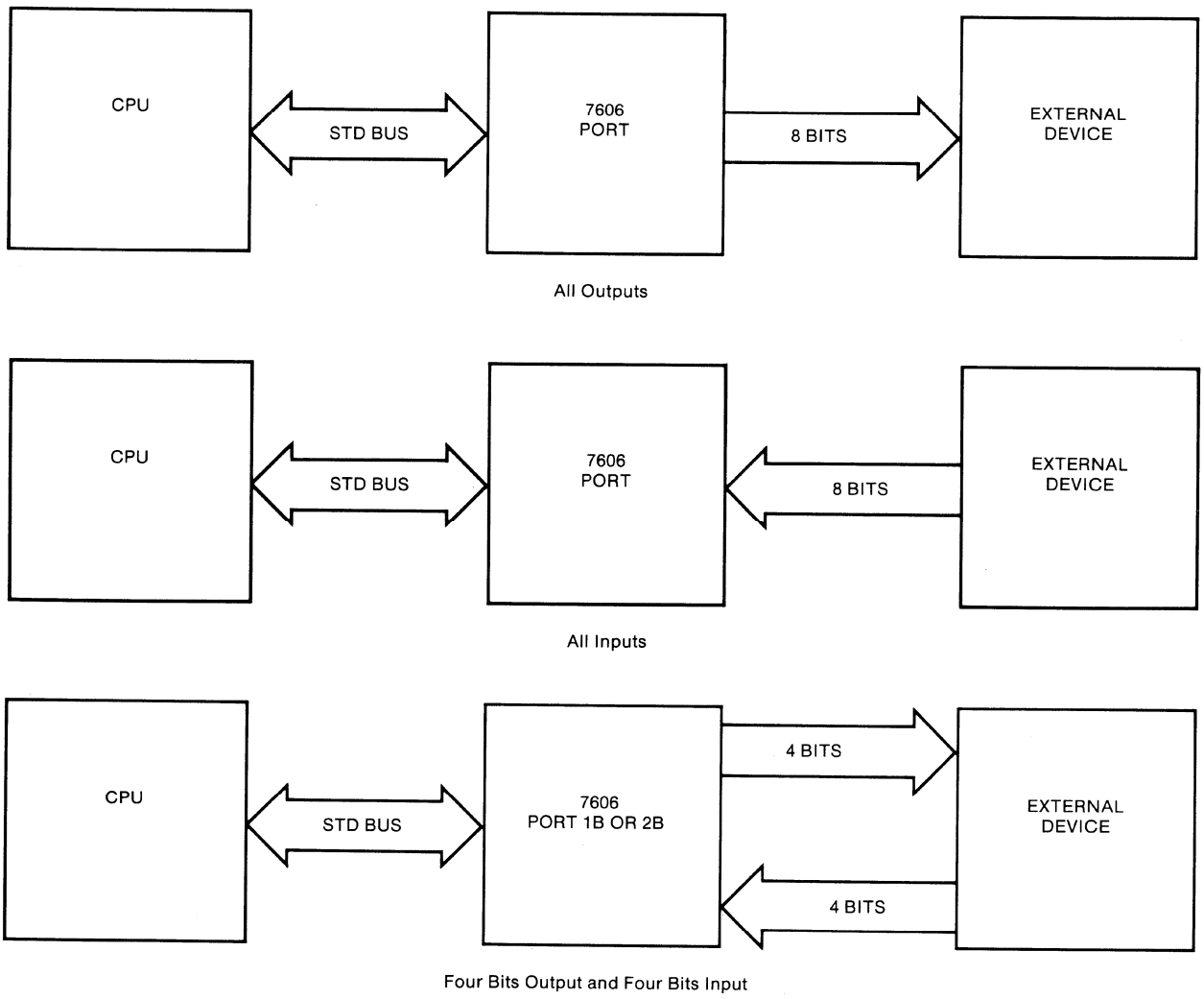
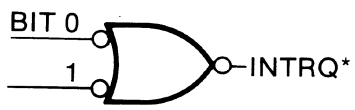


Figure 3-9. Examples of Mode 3 Operation (7606)

A second interrupt will not be generated until a change occurs in the status of the logical equation. As an example, assume that a port on the 7606 is programmed as follows:

- Mode 3 operation
- Interrupt enabled
- Logical OR function
- Active low
- Bits 2-7 masked



Under these conditions, if one unmasked line of the port data bus becomes active (Logical 0), an interrupt request is generated; if a second unmasked line becomes active (Logical 0) concurrently, a new interrupt request is **not** generated.

If the result of a logical operation becomes true immediately prior to or during an M1*, an interrupt request occurs after the trailing edge of M1*, provided the logical equation remains true after M1* returns high.

Figure 3-11 illustrates a port operating with the above programming sequence. All except bits 1A0* and 1A1* are masked and not monitored, creating a two-input negative logic OR gate. In the timing diagram, 1A0* is shown going low and creating an interrupt request. The CPU responds with an interrupt acknowledge cycle. The 7606 port with a pending interrupt request places its vector on the STD data bus. Program execution is transferred to an interrupt service routine. 1A0* is shown going inactive either by itself or as a result of action taken in the service routine. This action makes the logic equation false. An RETI instruction is then issued at the end of the service routine, clearing the interrupt request of the port under service. 1A1* then goes low, making the logical equation true and generating an interrupt request. This example illustrates two important points when dealing with interrupts for a port programmed in Mode 3 operation:

1. 1A1* must not go low before 1A0* goes high to make the logic equation false — a requirement for 1A1* to generate an interrupt request.
2. The second important point is that to generate the second interrupt 1A1* must go low after the interrupt service routine of 1A0* clears the 7606 interrupt structure with the RETI instruction.

Note that if port 1A is programmed to operate in Mode 2, port 1B must be programmed in Mode 3 and all bits masked. Under these conditions, port 1B cannot generate an interrupt and must be polled. The same relationship holds true for ports 2A and 2B.

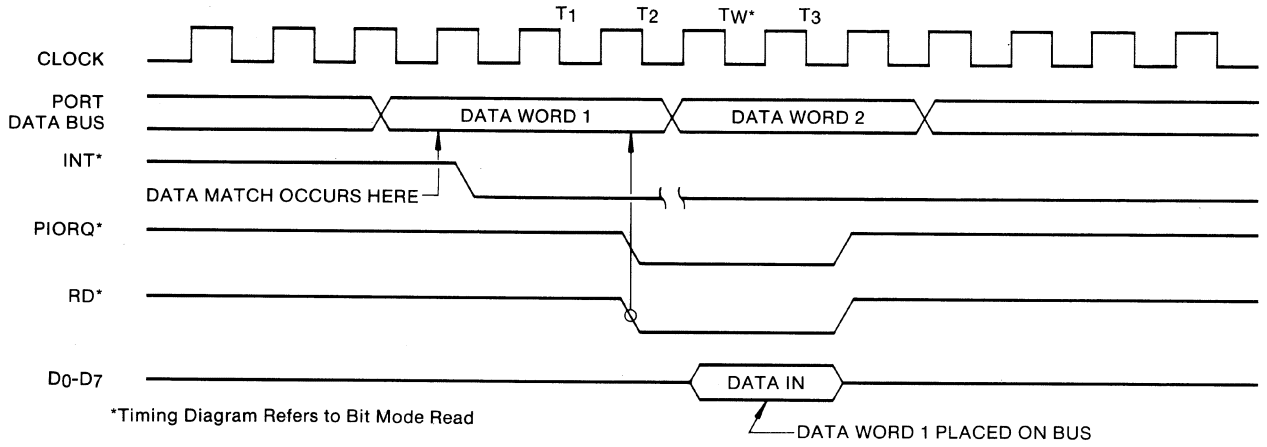


Figure 3-10. Port Timing for a CPU Read During Mode 3 (7606)

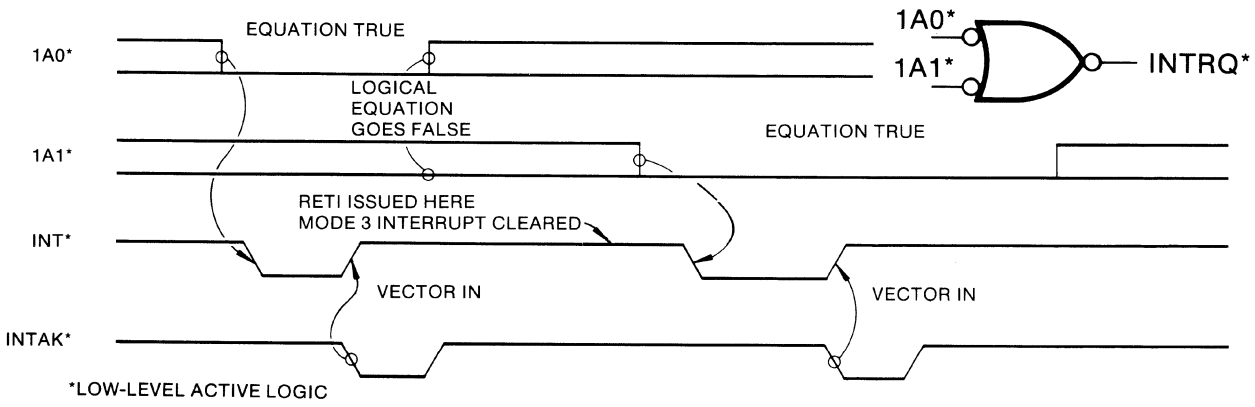


Figure 3-11. Port Timing for Mode 3 Example (7606)

PROGRAMMING

This section discusses the procedure for programming the 7606 in the modes of operation discussed previously in this section. Examples of the programming sequence for each mode are given in section 4.

If interrupts are to be used:

1. The Z80 must be programmed to operate under Mode-2 interrupts (not to be confused with Mode 2 operation of the 7606).
2. The Interrupt Register (IREG) of the Z80 must be loaded with the page address (most significant byte) of an indirect pointer to the interrupt service routine.

These two requirements are explained in greater detail in **Interrupt Structure** which follows in this section.

To program a 7606 port in Mode 0, 1, or 2 requires a control byte and an interrupt vector (if interrupts are enabled).

Mode Control Byte — selects the port operating mode (figure 3-12). This byte may be written at any time.

Interrupt Vector — Defines the line address (least significant byte) of an indirect pointer to the interrupt service routine. This byte is required by the Z80 Mode-2 operation). The interrupt vector is illustrated in figure 3-13.

To program a 7606 port in Mode 3 requires the Mode Control Byte, the Interrupt Vector (if interrupts are enabled), and two or three additional bytes as follows:

I/O Register Control Byte — When Mode 3 is selected the Mode Control Byte must be followed by another control byte defining which signals of the port data bus are to be input and which output. This must be done in 4-bit sections. The possibilities are:

- Bits 0-7 are all output
- Bits 0-7 are all input
- Bits 0-3 are input and 4-7 are output (ports 1B or 2B only)
- Bits 0-3 are output and 4-7 are input (ports 1B or 2B only)

The I/O Register Control Byte is illustrated in figure 3-14.

Interrupt Control Byte — In Mode 3 interrupt requests are generated as logic functions of the port data bus signal levels. The Interrupt Control Byte sets the logic conditions and the logic levels required for generating an interrupt. Two logic conditions or functions are available: AND (if all input bits change to the active level an interrupt request is generated), and OR (if any one of the input bits changes to the active level an interrupt request is generated). The active level of the input bits can also be set either high or low. The Interrupt Control Byte is illustrated in figure 3-15. The bits of the port data bus that are not to generate an interrupt can be masked by setting the appropriate bit of the Interrupt Control Byte, indicating that a Mask Control Byte is to follow.

Mask Control Byte — This byte allows any port data bus bit or combination of bits to be masked. The Mask Control Byte is illustrated in figure 3-16. This byte must immediately follow the Interrupt Control Byte.

Note that if port 1A (2A) is programmed to operate in Mode 2, port 1B (2B) must be programmed in Mode 3 and all bits masked. Under these conditions, port 1B (2B) cannot generate an interrupt and must be polled.

There is one other control byte that can be used to enable or disable the interrupts for each port on an individual basis:

Interrupt Disable Byte — This byte can be used without altering the Interrupt Control Byte. The Interrupt Disable Byte is illustrated in figure 3-17.

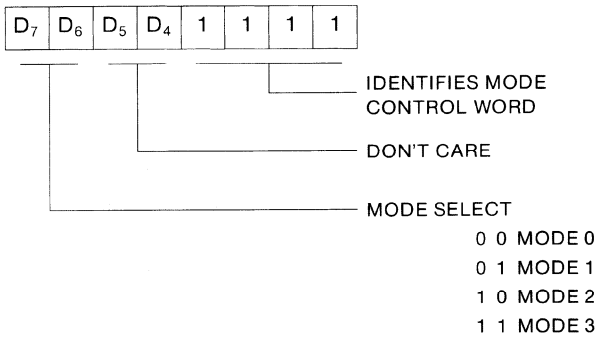


Figure 3-12. Mode Control Byte (7606)

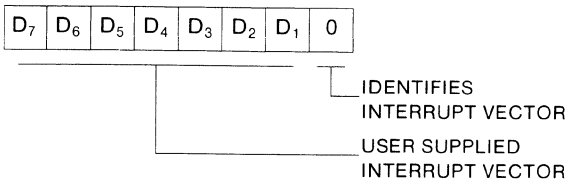


Figure 3-13. Interrupt Vector Byte (7606)

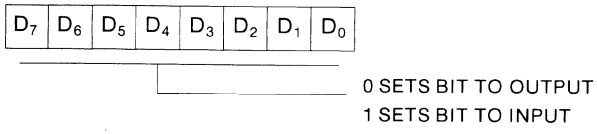
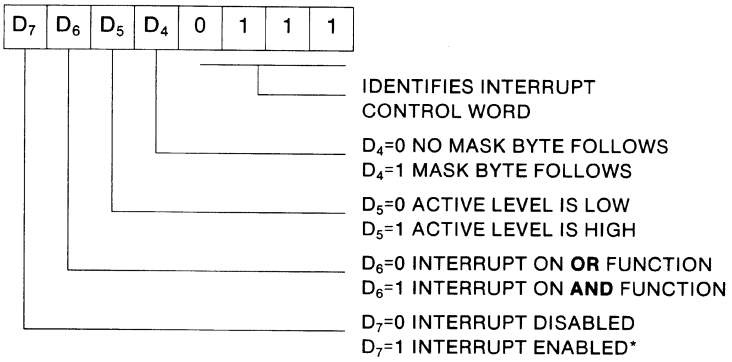


Figure 3-14. I/O Register Control Byte (7606)


D7 D6 D5 D4 0 1 1 1
IDENTIFIES INTERRUPT
CONTROL WORD
D4=0 NO MASK BYTE FOLLOWS
D4=1 MASK BYTE FOLLOWS
D5=0 ACTIVE LEVEL IS LOW
D5=1 ACTIVE LEVEL IS HIGH
D6=0 INTERRUPT ON OR FUNCTION
D6=1 INTERRUPT ON AND FUNCTION
D7=0 INTERRUPT DISABLED
D7=1 INTERRUPT ENABLED*

The port is not enabled until the interrupt enable is followed by an active M1.

Figure 3-15. Interrupt Control Byte (7606)

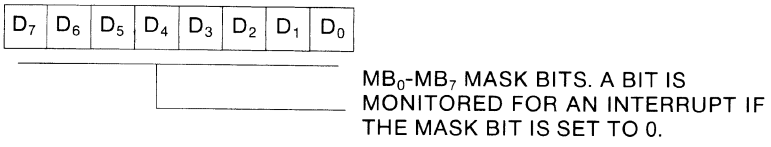


Figure 3-16. Mask Control Byte (7606)

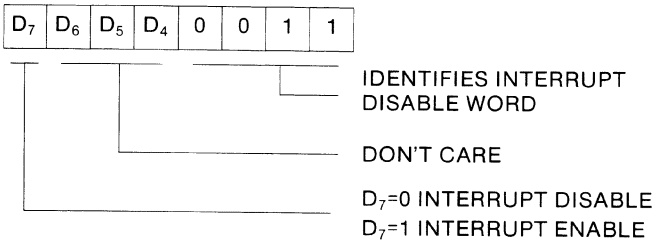


Figure 3-17. Interrupt Disable Byte (7606)

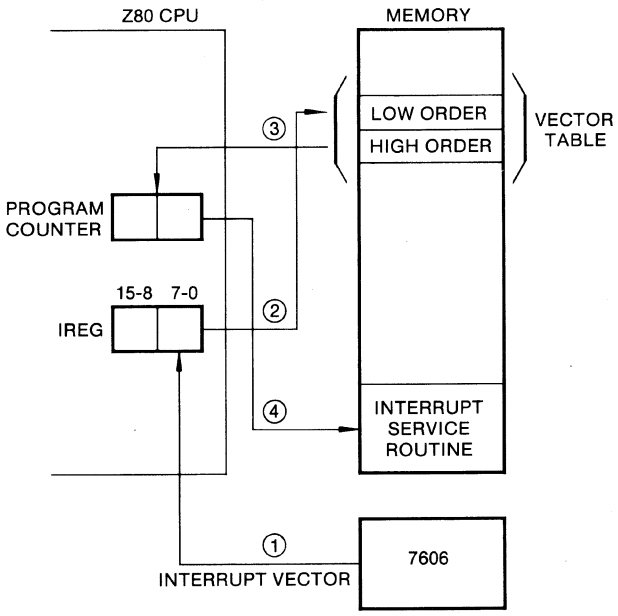
INTERRUPT STRUCTURE

Interrupts are a means of processing information in an asynchronous manner, providing an efficient interface between the CPU and a peripheral device. The interrupt causes the CPU to vector operation to a service routine designed for communication with the interrupting device. The communication generally involves the exchange of data (Mode 0, 1, and 2), or status and control information (Mode 3). After completion of the service routine, the CPU returns to the operation from which it was interrupted.

The 7606 supports the Z80 Mode-2 (not to be confused with Mode 2 operation of the 7606) interrupt structure. This mode allows each peripheral device to identify the starting address of the interrupt service routine. The 7606 also supports the daisy chain priority scheme which permits multiple peripherals to share the same interrupt request line.

Mode-2 is the most powerful of the Z80 interrupt structures. Figure 3-18 illustrates the sequence for processing with an interrupt. The CPU maintains an interrupt register (IREG) which is programmed with the page address (most significant byte) of an indirect pointer. The 7606 port generating an interrupt request (INTRQ*) places the line address (least significant byte) of the indirect pointer on the CPU data bus. This is done during an interrupt acknowledge (INTAK) cycle by the interrupting port meeting the priority chain requirements. The indirect pointer vectors the CPU to a vector table containing the address of the interrupt service routine.

The programmer maintains a vector table. The vector table includes a 16-bit starting address for each interrupt service routine. This table may be located anywhere in memory. When an interrupt is accepted, the CPU combines the contents of the IREG with the vector provided by the peripheral. This combination forms an address pointing to a location in the vector table. Specifically, that location and the one following are the starting address of the interrupt service routine.



NOTES:
1. Interrupt vector generated by 7606 is read by CPU during interrupt acknowledge cycle.
2. Vector combined with I register contents form 16-bit memory address pointing to vector table.
3. Two bytes are read sequentially from vector table. These two bytes are read into the program counter (PC).
4. Processor control is transferred to interrupt service routine and execution continues.

Figure 3-18. Vector Processing Sequence (7606)

The 7606 supports the daisy chain priority structure which assigns a priority to each port on a 7606 and to each 7606 in the STD BUS system. The daisy chain is implemented with three STD BUS signals: Priority Chain Input (PCI), Priority Chain Output (PCO), and Interrupt Request (INTRQ*).

The PCI and PCO lines provide for the establishment of priority among the ports capable of generating an interrupt. On the 7606 the priority order is 1A, 2A, 1B, and 2B. For multiple 7606 cards in an STD BUS system the priority is established as illustrated in figure 3-19. 7606 #1 has the highest priority because its PCI is not connected to a PCO of another card, and is permanently high. 7606 #2 has the lowest priority because its PCO is not connected to a PCI of another device. There can be no empty card slots in the daisy chain. If an STD card not using the daisy chain is placed between 7606 #1 and 7606 #2 the PCI must be connected directly to the PCO.

A typical nested interrupt sequence illustrating the operation of the daisy chain is shown in figure 3-20. This figure shows each port on the 7606 and the operation occurring when a lower priority device under service is interrupted by a device of higher priority. PCI and PCO are shown for each port to demonstrate the operation of the priority chain.

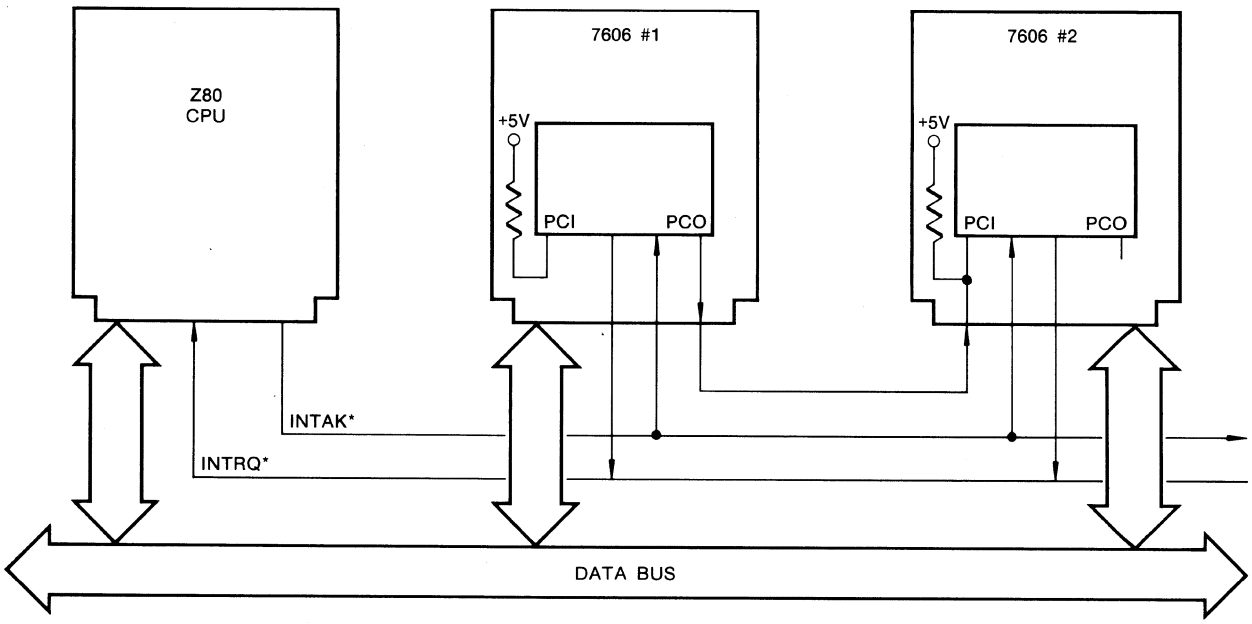


Figure 3-19. Serial Priority Scheme (7606)

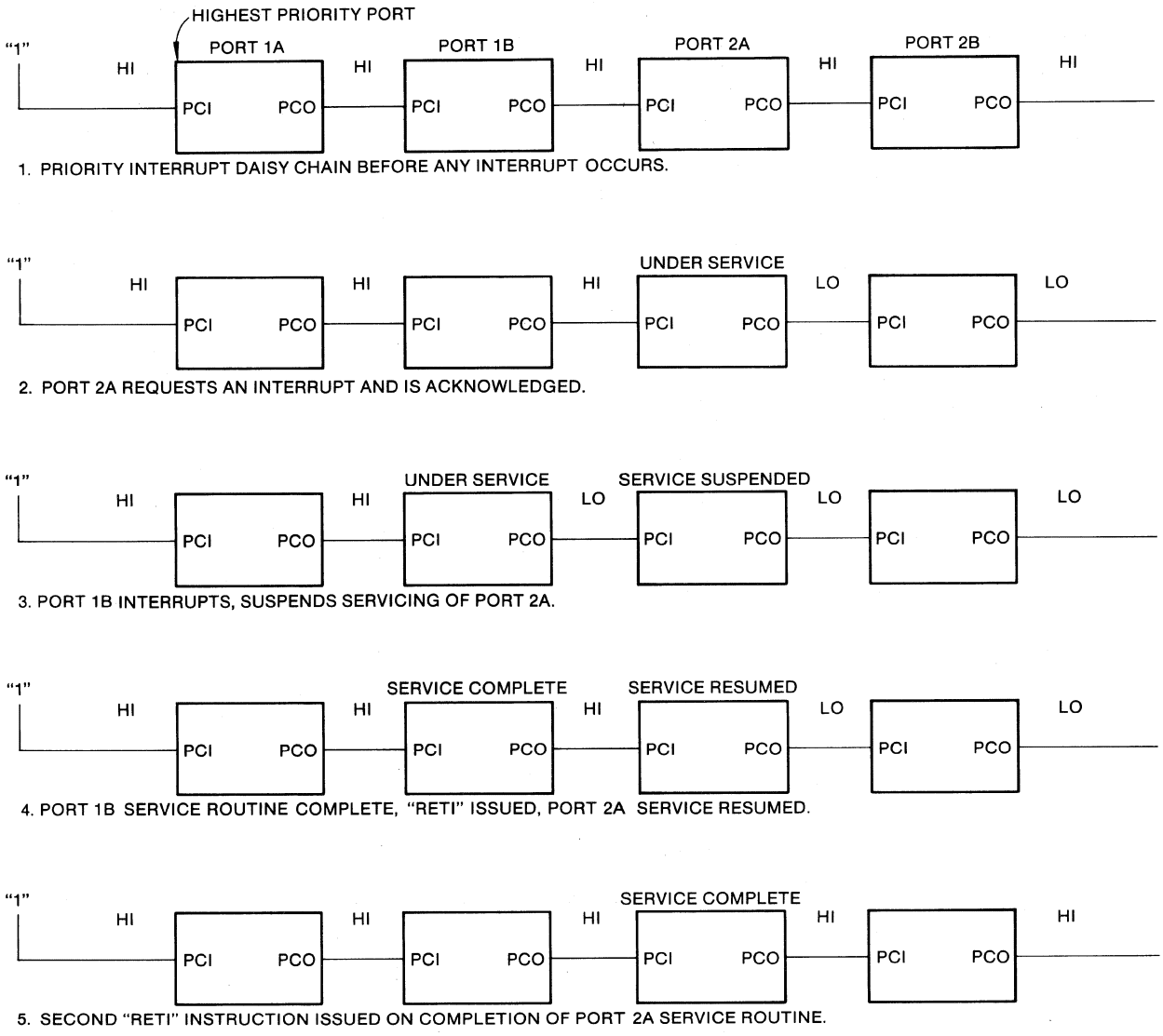


Figure 3-20. Daisy Chain Interrupt Servicing

The timing for the interrupt request/acknowledge cycle is illustrated in figure 3-21. The signals are those present on the STD BUS. For a detailed timing diagram refer to figure 2-6.

When a 7606 port has an interrupt pending it lowers the INTRQ* and the PCO signals. The active INTRQ* signal indicates to the CPU that an interrupt is pending. The inactive PCO signal inhibits all the ports of lower priority from responding to further interrupt operations. The active PCI signal ensures no devices of higher priority are requesting an interrupt.

The Z80 CPU samples the INTRQ* signal on the falling edge of CLOCK* during the last T state of the last machine cycle of any instruction. The CPU begins execution of an INTAK cycle if INTRQ* is active. The INTAK cycle is presented to the 7606 in two forms: The STD BUS INTAK* signal and the execution of an M1* cycle with IORQ* active. The INTAK* signal reverses the direction of the data bus buffers in anticipation of sending a vector to the CPU. As soon as M1* and IORQ* go active, the port with PCI high and PCO low gates an 8-bit vector onto the data bus. PCO remains low until a return from interrupt (RETI) instruction is executed while PCI is high. RETI is the last instruction in the interrupt service routine.

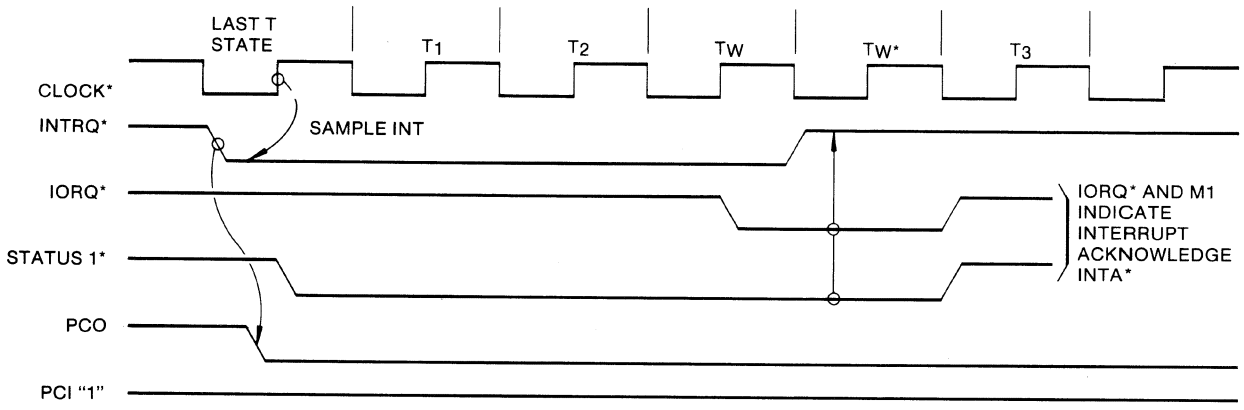


Figure 3-21. Interrupt Request/Acknowledge Timing (7606)

The timing for the return from interrupt cycle is illustrated in figure 3-22. For a detailed timing diagram refer to figure 2-6.

The CPU begins execution of the return from interrupt cycle with an RETI instruction. Each 7606 port monitors the STD data bus for the RETI instruction. RETI is a two-byte opcode ED-4D.

A port can request an interrupt at any time by activating INTRQ* and bringing PCO low. It is possible for a port whose interrupt is being serviced to have PCI low. (This would occur if a higher priority device requested an interrupt, but a lower priority device is being serviced and has not yet enabled interrupts.) To make sure the right port decodes the RETI instruction, all ports bring PCO high momentarily when the ED is seen during an ED-4D (RETI) instruction fetch. The device with an interrupt under service does not allow PCO to go high and resets itself when the 4D byte is fetched.

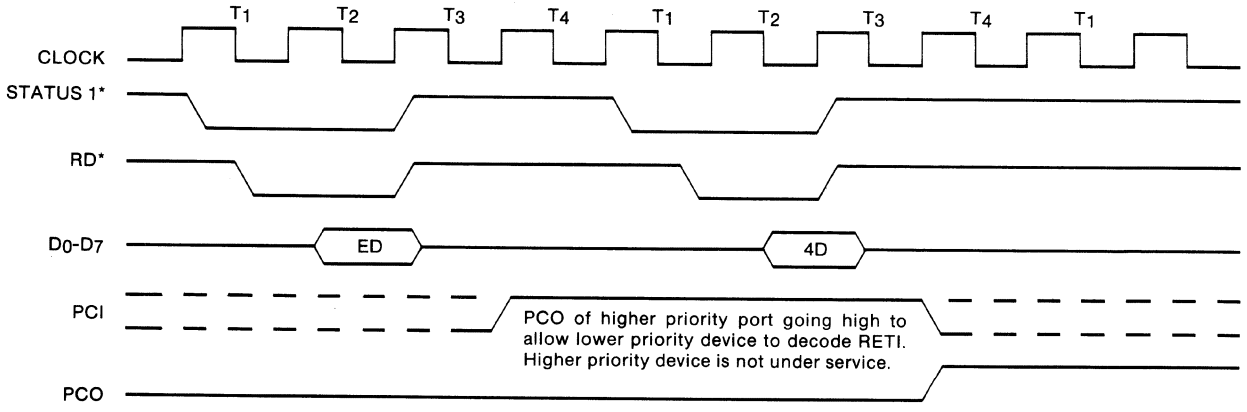


Figure 3-22. Return from Interrupt Timing for a Port Currently Under Service (7606)

SECTION 4
Operating Software

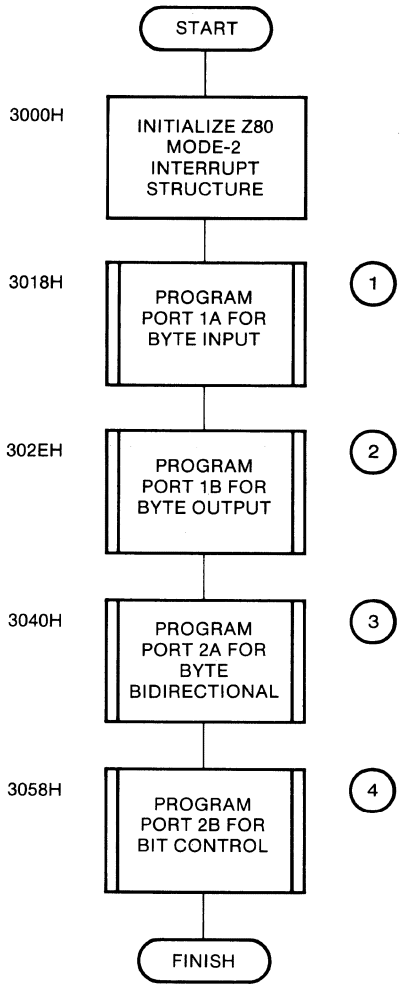
INTRODUCTION

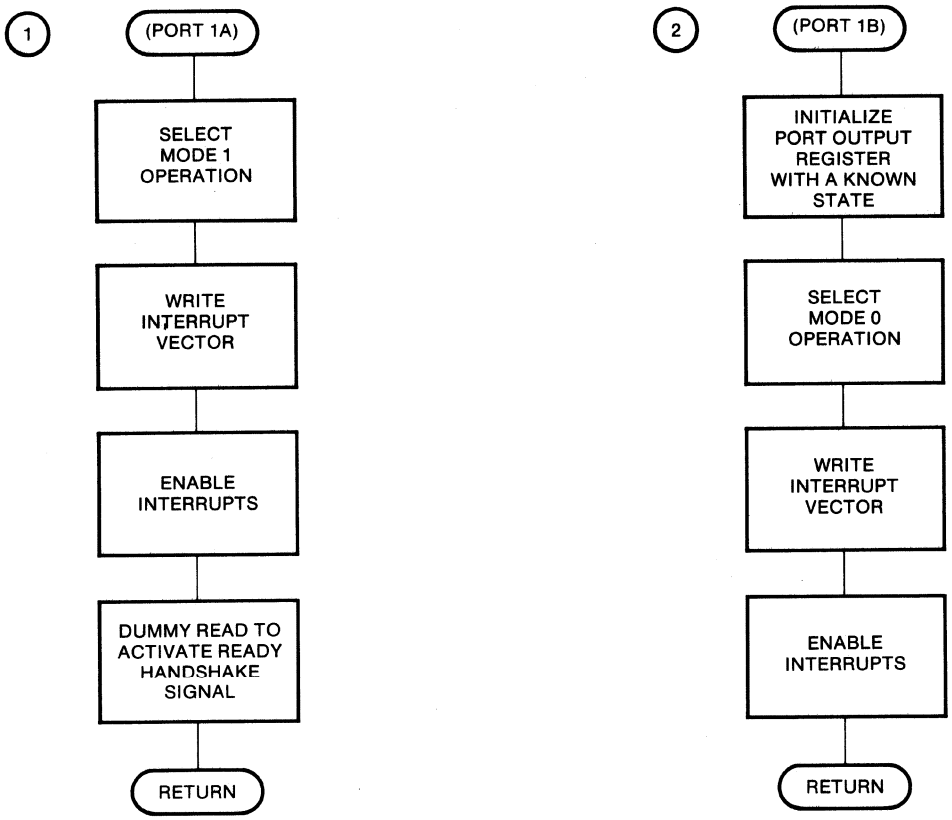
A 7606 port must be hardware configured and software programmed to operate in one of the four modes. The hardware configuration is discussed in section 2 and the software programming in section 3.

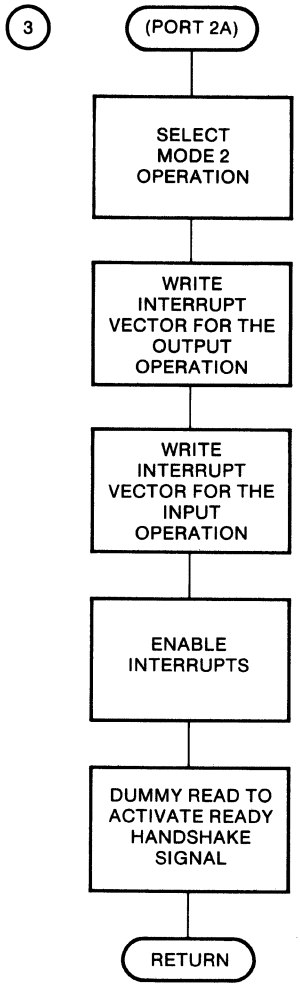
Section 4 presents the source code of an assembly language program that software configures a 7606 as illustrated in table 4-1. The port addresses used by this program are also listed in table 4-1 and represent the configuration of the 7606 as shipped. Table 4-1 shows port 2A operating in a bidirectional mode (Mode 2). This requires the use of the handshake signals and interrupt capabilities of port 2B. Port 2B is programmed with a mask control byte that prevents the bit Control Mode from generating interrupts. This requires that port 2B operate in a polled structure.

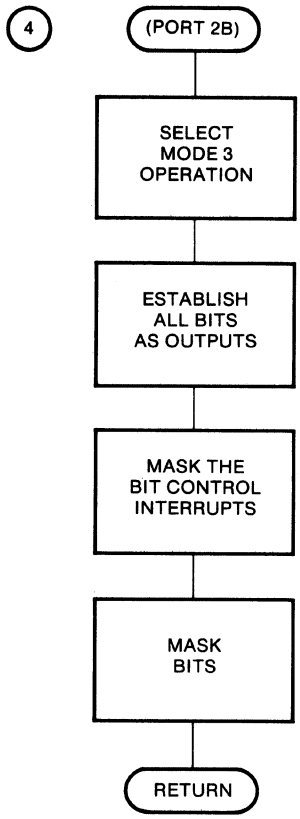
7606 PORT	PORT ADDRESS (HEX)	PROGRAMMED MODE OF OPERATION
1A	Data 08 Control 09	Byte Input (Mode 0)
1B	Data 0A Control 0B	Byte Output (Mode 1)
2A	Data 0C Control 0D	Byte Bidirectional (Mode 2)
2B	Data 0E Control 0F	Byte Control (Mode 3)

Table 4-1. Programmed Configuration for the Operating Software (7606)




1
(PORT 1A)
SELECT MODE 1 OPERATION
WRITE INTERRUPT VECTOR
ENABLE INTERRUPTS
DUMMY READ TO ACTIVATE READY HANDSHAKE SIGNAL
RETURN
2
(PORT 1B)
INITIALIZE PORT OUTPUT REGISTER WITH A KNOWN STATE
SELECT MODE 0 OPERATION
WRITE INTERRUPT VECTOR
ENABLE INTERRUPTS
RETURN


3
(PORT 2A)
SELECT MODE 2 OPERATION
WRITE INTERRUPT VECTOR FOR THE OUTPUT OPERATION
WRITE INTERRUPT VECTOR FOR THE INPUT OPERATION
ENABLE INTERRUPTS
DUMMY READ TO ACTIVATE READY HANDSHAKE SIGNAL
RETURN


4
(PORT 2B)
SELECT MODE 3 OPERATION
ESTABLISH ALL BITS AS OUTPUTS
MASK THE BIT CONTROL INTERRUPTS
MASK BITS
RETURN

```
ADRdata!     LABEL      !INSTR !    MODIFIER    !          COMMENTS
-------!---------------!------!----------------!----------------------------
3000 3E!(INITIALIZE)    !LDAI  !                !  LOAD THE PAGE ADDRESS (30H)
  01 30!                ! ---- !30              !INTO THE Z80 INTERRUPT
  02 ED!                !  ED  !                !REGISTER.
  03 47!                !  47  !                !
  04 ED!                !  ED  !                !  PROGRAM THE Z80 FOR MODE-2
  05 5E!                !  5E  !                !INTERRUPT STRUCTURE.
  06 CD!                !JS    !UNCONDITIONAL   !  INITIALIZE PORT 1A FOR BYTE
  07 18!                ! ---- !(PORT 1A)       !INPUT.
  08 30!                ! ---- !                !
  09 CD!                !JS    !UNCONDITIONAL   !  INITIALIZE PORT 1B FOR BYTE
  0A 2E!                ! ---- !(PORT 1B)       !OUTPUT.
  0B 30!                ! ---- !                !
  0C CD!                !JS    !UNCONDITIONAL   !  INITIALIZE PORT 2A FOR BYTE
  0D 40!                ! ---- !(PORT 2A)       !BIDIRECTIONAL.
  0E 30!                ! ---- !                !
  0F CD!                !JS    !UNCONDITIONAL   !  INITIALIZE PORT 2B FOR BIT
3010 58!                ! ---- !(PORT 2B)       !CONTROL.
  11 30!                ! ---- !                !
  12 FB!                !ENI   !                !  ENABLE INTERRUPTS.
  13 C9!   ----------   !RTS   !UNCONDITIONAL   !  RETURN.
  14 FF!                !      !                !
  15 FF!                !      !                !
  16 FF!                !      !                !
  17 FF!                !      !                !
3018 3E!(PORT 1A)       !LDAI  !                !  WRITE THE MODE CONTROL BYTE
  19 4F!                ! ---- !4F              !FOR MODE 1 TO THE CONTROL PORT.
  1A D3!                !OPA   !                !
  1B 09!                ! ---- !09              !
  1C 3E!                !LDAI  !                !  WRITE THE INTERRUPT VECTOR
  1D 72!                ! ---- !72              !BYTE TO THE CONTROL PORT.
  1E D3!                !OPA   !                !
  1F 09!                ! ---- !09              !
3020 3E!                !LDAI  !                !  WRITE THE INTERRUPT DISABLE
  21 83!                ! ---- !83              !BYTE TO THE CONTROL PORT TO
  22 D3!                !OPA   !                !ENABLE THE INTERRUPTS.
  23 09!                ! ---- !09              !
  24 DB!                !IPA   !                !  DUMMY INPUT TO RAISE THE
  25 08!                ! ---- !08              !1ARDY HANDSHAKE SIGNAL.
  26 C9!   ----------   !RTS   !UNCONDITIONAL   !  RETURN.
  27 FF!                !      !                !
  28 FF!                !      !                !
  29 FF!                !      !                !
  2A FF!                !      !                !
  2B FF!                !      !                !
  2C FF!                !      !                !
  2D FF!                !      !                !
302E 3E!(PORT 1B)       !LDAI  !                !  LOAD AN FF TO THE PORT. THIS
  2F FF!                ! ---- !FF              !ENSURES THAT THE PORT DATA BUS
3030 D3!                !OPA   !                !WILL COME UP IN A KNOWN STATE
  31 0A!                ! ---- !0A              !WHEN MODE 0 IS SELECTED.
  32 3E!                !LDAI  !                !  LOAD THE MODE CONTROL BYTE
  33 0F!                ! ---- !0F              !FOR MODE 0 TO THE CONTROL PORT.
  34 D3!                !OPA   !                !
  35 0B!                ! ---- !0B              !
  36 3E!                !LDAI  !                !  WRITE THE INTERRUPT VECTOR
  37 74!                ! ---- !74              !BYTE TO THE CONTROL PORT.
  38 D3!                !OPA   !                !
  39 0B!                ! ---- !0B              !
  3A 3E!                !LDAI  !                !  WRITE THE INTERRUPT DISABLE
  3B 83!                ! ---- !83              !BYTE TO THE CONTROL PORT TO
  3C D3!                !OPA   !                !ENABLE THE INTERRUPTS.
  3D 0B!                ! ---- !0B              !
  3E C9!   ----------   !RTS   !UNCONDITIONAL   !  RETURN.
  3F FF!                !      !                !
```

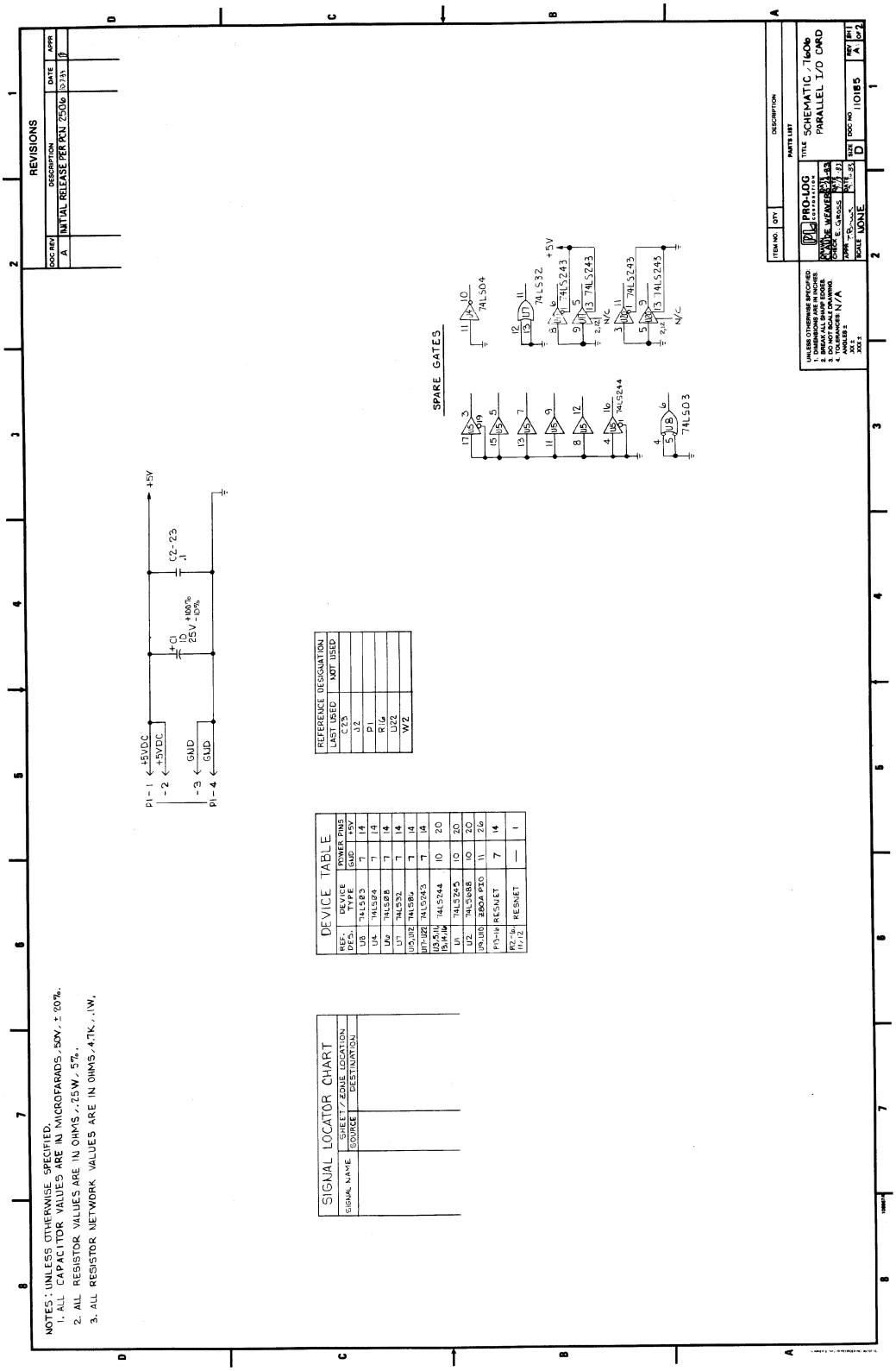
```
ADRdata!     LABEL      !INSTR !    MODIFIER    !          COMMENTS
-------!---------------!------!----------------!-----------------------------
3040 3E!(PORT 2A)       !LDAI  !                !  WRITE THE MODE CONTROL BYTE
  41 8F!                ! ---- !8F              !FOR MODE 2 TO THE CONTROL PORT.
  42 D3!                !OPA   !                !
  43 0D!                ! ---- !0D              !
  44 3E!                !LDAI  !                !  WRITE THE INTERRUPT VECTOR
  45 76!                ! ---- !76              !BYTE TO CONTROL PORT 2A FOR
  46 D3!                !OPA   !                !THE OUTPUT OPERATION.
  47 0D!                ! ---- !0D              !
  48 3E!                !LDAI  !                !  WRITE THE INTERRUPT VECTOR
  49 78!                ! ---- !78              !BYTE TO CONTROL PORT 2B FOR
  4A D3!                !OPA   !                !THE INPUT OPERATION.
  4B 0F!                ! ---- !0F              !
  4C 3E!                !LDAI  !                !  WRITE THE INTERRUPT DISABLE
  4D 83!                ! ---- !83              !BYTE TO THE CONTROL PORTS TO
  4E D3!                !OPA   !                !ENABLE THE INTERRUPTS.
  4F 0D!                ! ---- !0D              !
3050 D3!                !OPA   !                !
  51 0F!                ! ---- !0F              !
  52 DB!                !IPA   !                !  DUMMY INPUT TO RAISE THE
  53 0C!                ! ---- !0C              !2ARDY HANDSHAKE SIGNAL.
  54 C9!   ----------   !RTS   !UNCONDITIONAL   !  RETURN.
  55 FF!                !      !                !
  56 FF!                !      !                !
  57 FF!                !      !                !
3058 3E!(PORT 2B)       !LDAI  !                !  WRITE THE MODE CONTROL BYTE
  59 CF!                ! ---- !CF              !FOR MODE 3 TO THE CONTROL PORT.
  5A D3!                !OPA   !                !
  5B 0F!                ! ---- !0F              !
  5C 3E!                !LDAI  !                !  WRITE THE I/O REGISTER
  5D 00!                ! ---- !00              !CONTROL BYTE. THIS ASSIGNS ALL
  5E D3!                !OPA   !                !EIGHT BITS AS OUTPUTS.
  5F 0F!                ! ---- !0F              !
3060 3E!                !LDAI  !                !  WRITE THE INTERRUPT CONTROL
  61 97!                ! ---- !97              !BYTE TO THE CONTROL PORT.
  62 D3!                !OPA   !                !
  63 0F!                ! ---- !0F              !
  64 3E!                !LDAI  !                !  WRITE THE MASK CONTROL BYTE
  65 FF!                ! ---- !FF              !TO THE CONTROL PORT MASKING
  66 D3!                !OPA   !                !OFF ALL BITS. THIS PERMITS THE
  67 0F!                ! ---- !0F              !PORT 2B INTERRUPTING CAPABILITY
  68 C9!   ----------   !RTS   !UNCONDITIONAL   !TO BE USED BY PORT 2A. RETURN.
  69 FF!                !      !                !
  6A FF!                !      !                !
  6B FF!                !      !                !
  6C FF!                !      !                !
  6D FF!                !      !                !
  6E FF!                !      !                !
  6F FF!                !      !                !
3070 FF!                !      !                !
  71 FF!                !      !                !
3072 80!PORT 1A VECTOR  !  80  !                !  THESE ARE THE ADDRESSES OF
  73 30!                !  30  !                !THE INTERRUPT SERVICE ROUTINES
3074 90!PORT 1B VECTOR  !  90  !                !FOR EACH PORT ON THE 7606 CARD.
  75 30!                !  30  !                !FOR EXAMPLE: THE INTERRUPT
3076 A0!PORT 2A VECTOR  !  A0  !                !SERVICE ROUTINE FOR PORT 1A IS
  77 30!                !  30  !                !LOCATED AT 3000H.
3078 B0!PORT 2B VECTOR  !  B0  !                !
  79 30!                !  30  !                !
  7A FF!                !      !                !
  7B FF!                !      !                !
  7C FF!                !      !                !
  7D FF!                !      !                !
  7E FF!                !      !                !
  7F FF!                !      !                !
```

SECTION 5
Maintenance

REFERENCE DRAWINGS

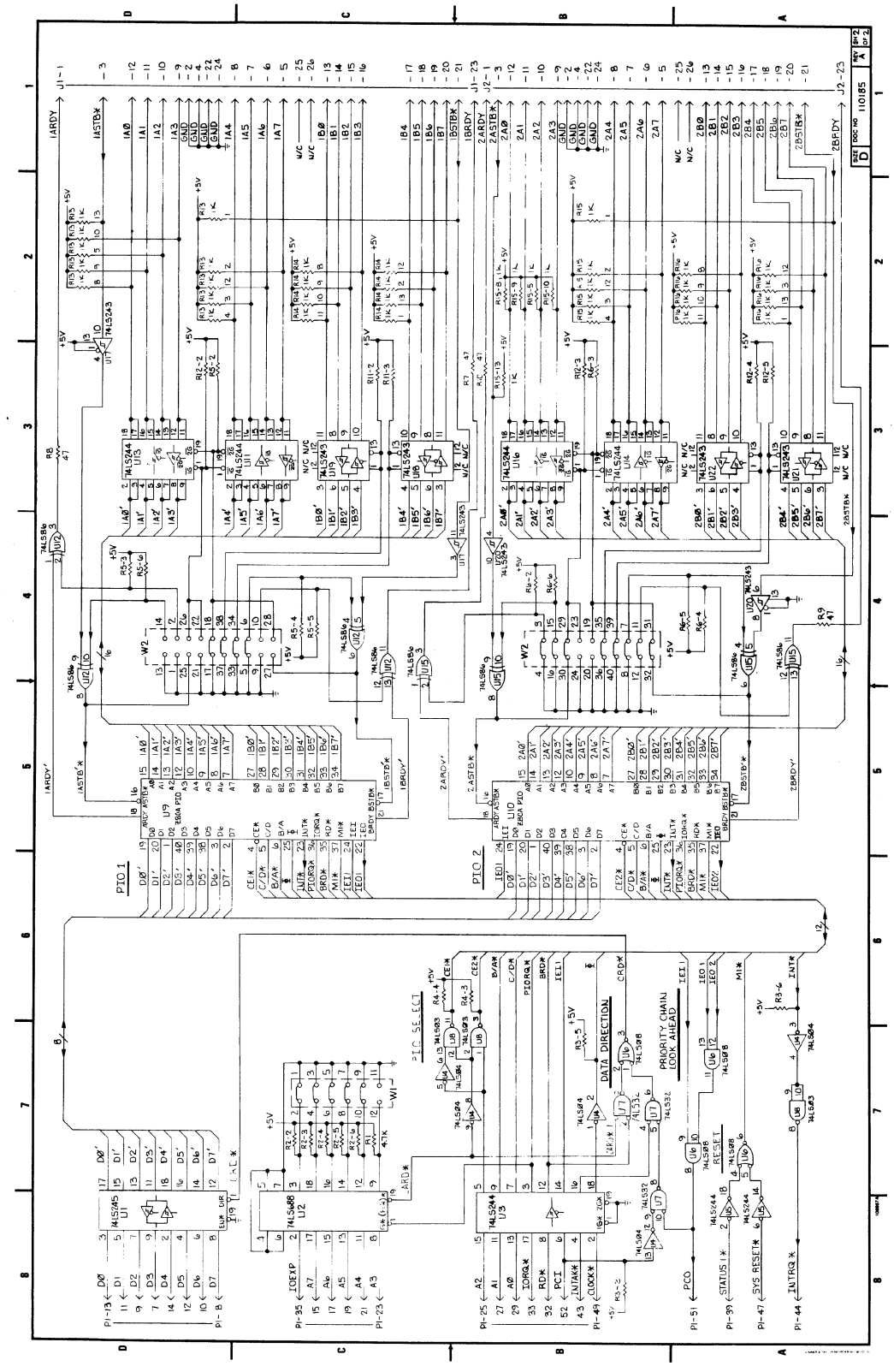
The schematic (figure 5-1) and assembly drawing (figure 5-2) in the following pages are included in this manual FOR REFERENCE USE ONLY. Because improvements in the product may have been made after this manual was published they may differ in some respects from the card and documentation that the user received from Pro-Log.

The schematic and assembly drawing shipped by Pro-Log with the card are those from which the card was manufactured.



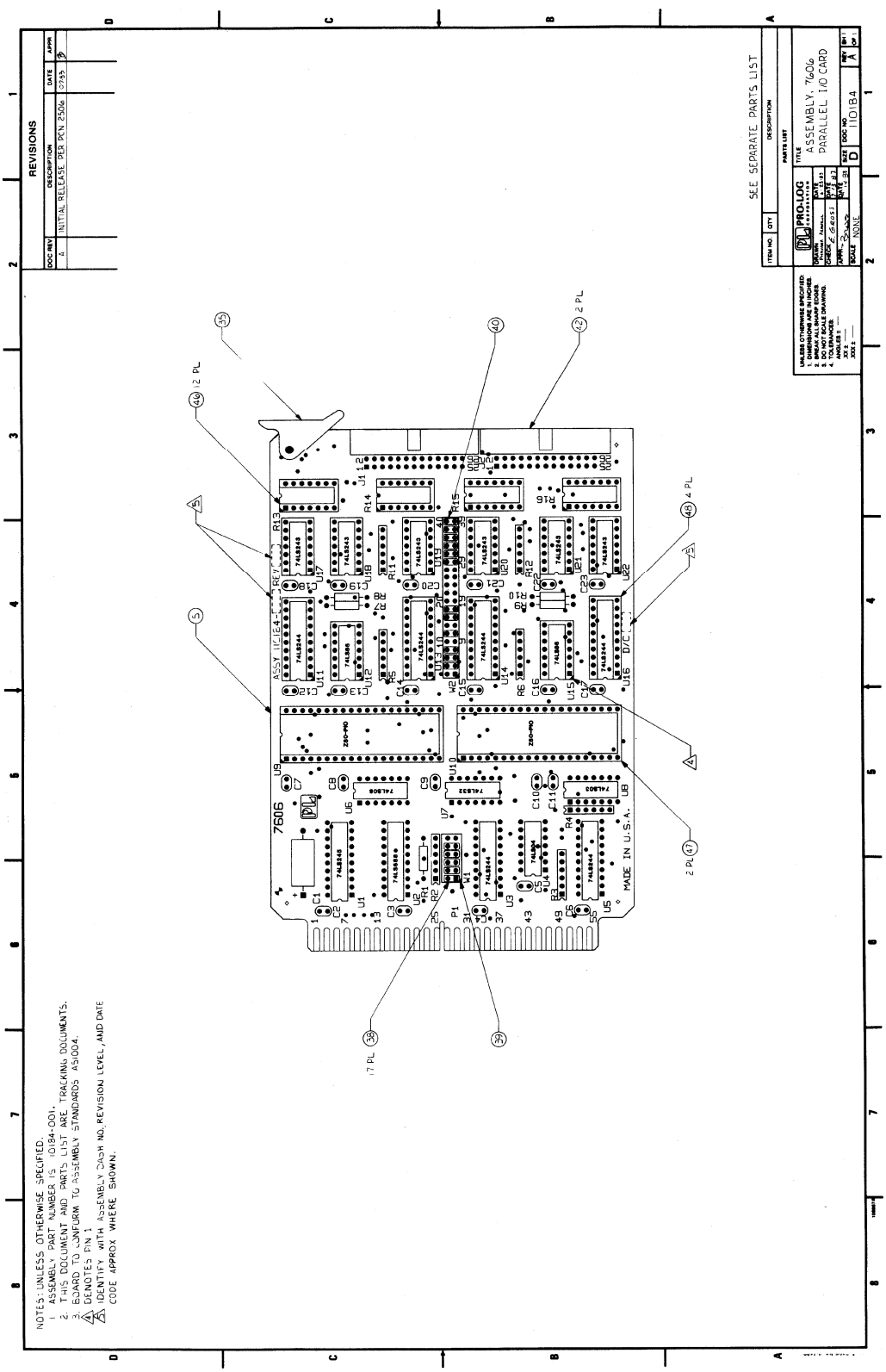
(FOR REFERENCE ONLY)

Figure 5-1. Schematic, Page 1 (7606)



(FOR REFERENCE ONLY)

Figure 5-2. Schematic, Page 2 (7606)



(FOR REFERENCE ONLY)

Figure 5-3. Assembly Drawing (7606)

RETURN FOR REPAIR PROCEDURES

Domestic Customers:

1. Call Customer Support at (800) 538-9570, ext. 3617. In California call (408) 646-3617 from 7 a.m. to 3 p.m. (PST).
2. Explain the problem and we may be able to solve it on the phone. If not, we will give you a Customer Return Order (CRO) number.

   Mark the CRO number on the shipping label, packing slip, and other paperwork accompanying the return. We cannot accept returns without a CRO.
3. Please be sure to enclose a packing slip with CRO number, serial number of the equipment, if applicable, and reason for return, with the name and telephone number of the person we should contact (preferably the user), if we have any further questions.
4. Pack the equipment in a solid cardboard box secured with packing material.

CAUTION

Loose MOS integrated circuits, or any product containing CMOS integrated circuits, must be protected from electrostatic discharge during shipment. Use conductive foam pads or conductive plastic bags, and never place MOS or CMOS circuitry in contact with Styrofoam materials.

5. Ship prepaid and insured to:

   Pro-Log Corporation
   2411 Garden Road
   Monterey, California 93940

   Reference CRO #_____

International Customers:

Equipment repair is handled by your local Pro-Log distributor. If you need to contact Pro-Log, the factory can be reached at any time by TWX at 910-360-7082.

Limited Warranty

Limited Warranty: Seller warrants that the articles furnished hereunder are free from defects in material and workmanship and perform to applicable, published Pro-Log specifications for two years from date of shipment. This warranty is in lieu of any other warranty expressed or implied. In no event will Seller be liable for special or consequential damages as a result of any alleged breach of this warranty provision. The liability of Seller hereunder shall be limited to replacing or repairing, at its option, any defective units which are returned f.o.b. Seller's plant. Equipment or parts which have been subject to abuse, misuse, accident, alteration, neglect, unauthorized repair or installation are not covered by warranty. Seller shall have the right of final determination as to the existence and cause of defect. As to items repaired or replaced, the warranty shall continue in effect for the remainder of the warranty period, or for ninety (90) days following date of shipment by Seller or the repaired or replaced part whichever period is longer. No liability is assumed for expendable items such as lamps and fuses. No warranty is made with respect to custom equipment or products produced to Buyer's specifications except as specifically stated in writing by Seller and contained in the contract.

APPENDIX A
Application

BIT CONTROL (Mode 3)

Bit control is useful for monitoring and controlling an industrial process. Any of the four ports on the 7606 can be operated in the bit control mode.

Figure A-1 shows an application of the bit control mode in which a 7606 port is used to provide closed loop monitor and control of an oven. Suppose the oven became too hot. The following sequence of events illustrates the closed loop control of the problem.

A. The industrial process activates the "too hot" signal.

B. The "too hot" signal causes the 7606 port to interrupt the processor.

C. The processor vectors to the service routine for the interrupting port.

D. The service routine inputs the 7606 port and determines that it was the "too hot" signal that generated the interrupt.

E. The service routine then writes a new logical equation to the 7606 port that generates an interrupt when the "too hot" signal becomes inactive.

F. The service routine then activates the "decrease heat" signal and returns the CPU to the operation from which it was interrupted.

G. The industrial process cools the oven. When the temperature is within operating range, the "too hot" signal is made inactive.

H. The transition of the "too hot" signal causes the 7606 port to interrupt the processor.

I. The processor vectors to the service routine for the interrupting port.

J. The service routine inputs the 7606 port and determines that it is the removal of the "too hot" signal that generated an interrupt.

K. The service routine then places the "decrease heat" signal in its inactive state and returns the CPU to the operation from which it was interrupted.

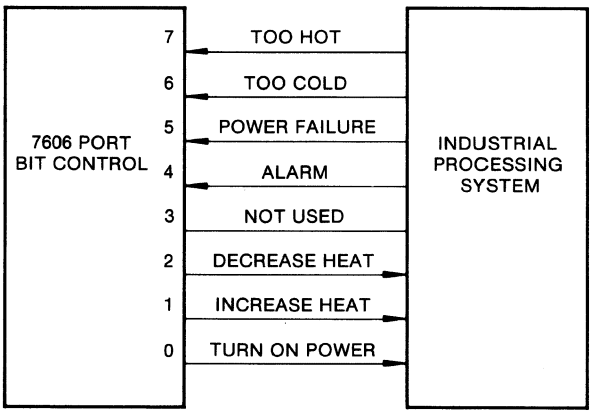


Figure A-1. Example Using Bit Control (Mode 3 7606)

The programming required for this operation is now illustrated. Port 1A is arbitrarily chosen for the Mode 3 operation. Port 1A must be hardware configured to have the upper four lines of the port data bus as inputs and the lower four as outputs.

Before port 1A can operate in Mode 3, the following programming sequence must be executed by the CPU.

1. Software select port 1A for Mode 3 operation. To do this, the control byte of figure A-2 must be output to port 1A by the CPU. See figure 3-12 for the format of the Mode Control Byte.

1	1	X	X	1	1	1	1

X = Don't Care

Figure A-2. Mode Control Byte (7606)

2. Whenever Mode 3 is selected, the next control word sent to the port must be the I/O Register Control Byte. For the example, the upper four bits are inputs and the lower four bits outputs. Figure A-3 illustrates the I/O Register Control Byte that the CPU must output to port 1A. See figure 3-14 for the format of the I/O Register Control Byte.

1	1	1	1	0	0	0	0

Figure A-3. I/O Register Control Byte (7606)

3. The example requires the CPU be interrupted when the "too hot" signal changes state. The address of the service routine must be placed in a vector table. The address of the vector table location containing the service routine address must be programmed. The upper byte must be loaded into the Z80 IREG and the lower byte into port 1A on the 7606. Figure A-4 illustrates the vector format the CPU must output to port 1A. See section 3 for more information.

V_7	V_6	V_5	V_4	V_3	V_2	V_1	0

Figure A-4. Vector Control Byte (7606)

4. The Interrupt Control Byte is sent next to port 1A by the CPU. The Interrupt Control Byte enables the interrupts (Bit 7), enables the logical OR function (Bit 6), establishes the active high level of the signals (Bit 5), and indicates that the next byte sent to port 1A will be a mask byte (Bit 4). Figure A-5 illustrates the Interrupt Control Byte to be output by the CPU. See figure 3-15 for the format of the Interrupt Control Byte.

1	0	1	1	0	1	1	1

Figure A-5. Interrupt Control Byte (7606)

5. The Mask Control Byte must be sent next to port 1A by the CPU. The Mask Control Byte is illustrated in figure A-6. For the example, Bits 4 through 7 are not masked. See figure 3-16 for the format of the Mask Control Byte.

0	0	0	0	1	1	1	1

Figure A-6. Mask Control Byte (7606)

Steps 4 and 5 establish a logical equation for port 1A. An interrupt is generated if the logical equation becomes true, e.g., if any one (logical OR) of the inputs (unmasked signals) goes high (active state), an interrupt is generated. Such is the case when the "too hot" signal goes high in step 1 of the closed loop description.

Step E of the closed loop description specifies creating a new logic equation to generate an interrupt when the "too hot" signal becomes inactive. To do this, the service routine must write a new Interrupt Control byte to port 1A. The new Interrupt Control byte is illustrated in figure A-7. It differs from the previous Interrupt Control byte in that the logical function (Bit 6) is set for AND, the active level (Bit 5) is set low. The interrupt mask need not be changed (Bit 4). See figure 3-15 for the format of the Interrupt Control Byte.

Now when the "too hot" signal goes low (step F), an interrupt is once again generated.

1	1	0	0	0	1	1	1

Figure A-7. New Interrupt Control Byte (7606)

(CUT HERE)

Yes, I'm interested in Pro-Log.

Please send me more information about your:

- ☐ 7000 Series STD BUS Cards
- ☐ STD BUS Packaging Solutions
- ☐ STD BUS Subsystem
- ☐ STD BUS Systems
- ☐ Designer Software
- ☐ Design Courses and Seminars
- ☐ Programmer Product Line
- ☐ Other _____

- ☐ I am currently using Pro-Log equipment.
- ☐ Add my name to your mailing list.
- ☐ Please have a Sales Representative contact me.

FOLD

Reader Comments: _____

FOLD

Name _____ City _____

Company _____ State _____ Zip _____

Address _____ Telephone _____

PLACE
POSTAGE
HERE

PRO-LOG®
CORPORATION
2411 GARDEN ROAD
MONTEREY, CA 93940

Attn: Literature Department